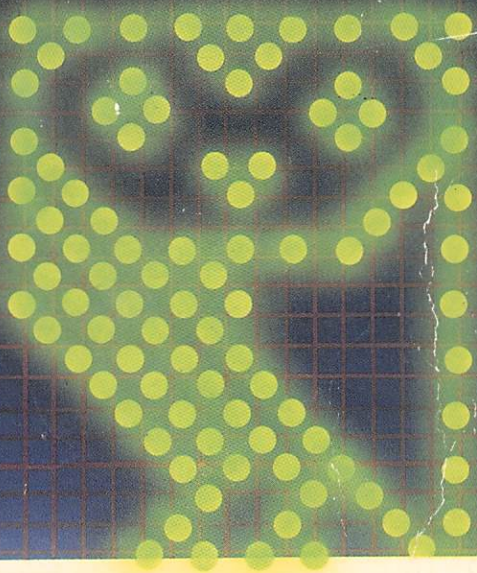


BBC PROGRAMS Volume 1



PROGRAMS BY CARL GRAHAM
EDITED BY NICK HAMPSHIRE

CONTENTS

Introduction	4
Introduction Notes	6
Star Trek	7
Space Blaster	23
Brick Basher	34
Space Invaders	42
Animals	49
Battleships	55
Mutant	72
On the Way To The Interview	93
Data Program	106
Character Editor	114
Split	119
Curve Fitting	122
Mr Ugly	130
Mr Ugly 80	133
Sketching (needs joystick)	136
Effects Program	140
Mummy	146
3D Graph	151
Recursive Pattern	153
Key Buffer	155
Hilberts Curves	157
Input 0-9	159
Sound	161
Four Pattern Generators	163
Digit Clock	165
Osfile	169
Fill Demo	171
Index	175

INTRODUCTION

This book is the first containing programs written exclusively by Mr Carl Graham, a long time user of the B.B.C. micro computer. All the programs are completely original and make excellent use of the machine's capabilities. I have enjoyed editing this book and am proud to be the first to introduce this new expert programmer to the industry.

NICK HAMPSHIRE

B.B.C. PROGRAMS

Introduction Notes.

When typing in programs, there are some important points to remember. If these instructions are not followed, you will either lose valuable computer memory or the programs will not work.

1. Some of the listings given, were completed using LIST07, so there are spaces just after the line number on each line. In any program, do not put spaces between the line numbers and at the start of the basic statement, as this wastes a very large amount of memory in long programs.
2. For the same reason, when typing in, or editing lines, be very careful not to type or to copy extra spaces on to the end of the line. This is very easily done by holding down the copy key too long.
3. In some programs the pound sign is used as a variable name, but on the listing, it is shown as "`^`". Note the difference between "`^`" and "`^`" which is SHIFT 7.
4. BBC basic does require spaces in some basic statements, one is sometimes needed between a variable name and a "`THEN`".
5. All programs will also run on a standard BBC B disk system if the above notes are followed.
6. The programs in this book have been tested on both versions of the BBC Basic, but not on OS 0.1. Programs that require the new versions of the operating system have been kept to a minimum.
7. Blank lines are entered by just typing one space after the line number.

STAR TREK

DESCRIPTION

The captain is on holiday and war with the Klingons has just broken out in sector 3. So, off you go with the rest of the crew of the Star Ship Enterprise to fight the demonic Klingons. Your reactions must be fast, because the Klingons don't wait for anyone and have a disconcerting habit of exterminating sluggish star ships. In this war you will be struggling for your survival.

RUNNING THE PROGRAM

When the game is run, it asks for the difficulty level, '1' is easy and '9' is virtually impossible for a beginner. After the message "ALL SYSTEMS GO" the screen will display a real time short range scan of the quadrant of the galaxy you are in. To the right of this, is your information panel showing fuel levels, current position, shield levels, etc. Below this the message "SHIELD FAILURE" is displayed if the shield has been drained by continuous Klingon bombardment. The bottom part of the screen is reserved for your commands.

The battle field is a galaxy 10 by 10 quadrants in size, each quadrant also being 10 by 10. The number of Klingons you have to destroy and the number of star bases for refuelling are set by the difficulty level at the start of the game. You have the following commands available:

F: Fire photon torpedoes. You will be told the number of torpedo banks that remain unfired and then asked for a direction (see below). This will be repeated until all banks have been fired or you type "S".

M: Move Enterprise. You are asked for a heading (see below).

L: The long range scanner is activated until another key is pressed. It shows the amounts of stars, star bases and Klingons in the surrounding quadrants, under the heading "S", "B" and "K" respectively.

S: Asks for new value of shield power. If the number of Klingons is high, more shield power is required and this is deducted from the fuel level each second.

R: Repair damage to the Enterprise. Once started this cannot be stopped until the repairs have been completed. While you are having repairs you are vulnerable to attack.

While commands that ask for more information are being typed, the display will freeze. When you press return on a line of data, for example a heading, the program will update the display to show what was happening while you were typing. When it has finished your command will be completed. So you may have been killed while you were typing and did not realise until you pressed return.

E: Tests to see if you have killed all the Klingons and it is the end of the game.

P: Panic Button activates emergency warp drive.

DIRECTIONS AND DISTANCES

A distance of 0.1 will move you to the next position horizontally or vertically. Also a distance of 1 will move you a quadrant width horizontally or vertically. Headings are given in degrees, for instance 0 is down, 90 is right, -90 is left and 180 is going up. Any angle between these can be used.

180

-90 - * - 90

0

Hint: If you get swamped by Klingons try shooting a star!

```

10MODE4
20PRINT"          STARTREK BY C.GRAHAM"
25
30REM{OS 1.0 ONWARDS ONLY}Debugging routine.
35ONERRORGOTO4320
36
37REM Galaxy quadrant & damage arrays
38
50DIM GZ(9,9),QZ(9,9),DZ(3)
52
55REM Set damage to none
56
60GOSUB1550
70ENVELOPE1,138,1,0,-1,10,7,10,70,-2,-1,-9,126
,100
74
75REM Define char. for Enterprose Klingons etc
.
76
80E=224
90VDU23,E,16,56,56,84,124,84,84,68
100AK=225
110VDU23,AK,130,124,56,16,16,16,56,16
120PHO=226
130VDU23,PHO,0,0,40,0,40,0,0,0
140AS=42
150AB=227
160VDU23,AB,60,66,165,153,153,165,66,60
170SUP=228
180VDU23,228,126,255,255,255,255,255,126
190EX=229
200VDU23,EX,189,126,255,255,255,255,126,189
210PRINTTAB(9,5);"S T A R T R E K "
220PRINTSTRING$(20,"*+")
230PRINT" You are the Captain of the star-ship
ENTERPRISE"
240PRINT" It is your mission to destroy all the
"/"KLINGONS before they DESTROY 'YOU'."
250PRINT"And plunder the galaxy."
260INPUT"HOW DIFFICULT DO YOU WANT THE MISSION"
"/" 1-9",IDZ
270IFIDZ>9THENPRINT"THAT'S IMPOSSIBLE":GOTO260

```

```

280IFIDZ<1THENPRINT" You could always play snap
":60T0260
290NBZ=15-RND(IDZ)
300NSZ=600+RND(100)
310CLS
320PRINT""
330NKZ=IDZ^3-RND(IDZ^2)+10
340PRINT""ENTERPRISE ENTERING GALAXY"
350FORIZ=1TONSZ
360LXZ=RND(10)-1
370LYZ=RND(10)-1
380GZ(LXZ,LYZ)=GZ(LXZ,LYZ)+1
390NEXT
400PRINT"WARNING ";NKZ;"KLINGONS INVADING GALAX
Y"
410FORIZ=1TONKZ
420LXZ=RND(10)-1
430LYZ=RND(10)-1
440GZ(LXZ,LYZ)=GZ(LXZ,LYZ)+10000
450NEXT
460PRINT;NBZ;" STARBASES TAKING UP POSITIONS"
470FORIZ=1TONBZ
480LXZ=RND(10)-1
490LYZ=RND(10)-1
500GZ(LXZ,LYZ)=GZ(LXZ,LYZ)+100
510NEXT
520PRINT"" ALL SYSTEMS GO"
530FORI=0T03000
540NEXT
550FL=10000
560PT=10
570SR=20
580TIME=0
590XE=RND(9)-1+RND(1)
600YE=RND(9)-1+RND(1)
605
610REM**** GALAXY FINISHED
615
620GOSUB 1620
630CLS
635
640REM=== SET UP QUADRANT

```



```

645
650SOUND1,-15,120,10
660VDU19,1,6;0;
670GCOL0,129
680CLG
690GCOL0,128
700GCOL0,0
710MOVE23,1000
720DRAW360,1000
730DRAW360,662
740DRAW23,662
750DRAW23,1000
760MOVE23,23
770DRAW1256,23
780DRAW1256,550
790DRAW23,550
800DRAW23,23
805
810REM==== DRAW QUADRANT
815
820FORIX=0TO9
830FORJZ=0TO9
840PRINTTAB(IX+1,JZ+1);
850QZ=QZ(IX,JZ)
860IFQZ=0THEN920
870IFQZ=1THENVDUAS:GOTO930
880IFQZ=100THENVDUAB:GOTO930
890IFQZ=10000THENVDUAK:GOTO930
900IFQZ=-1THENVDUBH:GOTO930
910IFQZ=-10THENVVDUE:GOTO930
920VDU32
930NEXT
940NEXT
950GOSUB980
960GOTO1150
965
970REM===== DATA DISPLAY
975
980COLOUR0
990COLOUR129
1000PRINTTAB(13,1)"CURRENT POSITION"

```

```

1010PRINTTAB(13,2)"[";LEFT$(STR$XE+"      ",8);"]
; [";LEFT$(STR$YE+"      ",8);"]
1020PRINTTAB(13,4);"ENERGY LEFT :";FL;"
"
1030PRINTTAB(13,6);"SHIELD POWER :";SR;"      "
1040PRINTTAB(13,8);"PHOTON TORP.S :";PT;"      "
1050T=TIME
1060PRINTTAB(13,10)"MISSION TIME :";TIME DIV100
1070PRINTTAB(1,12)"DAMAGE % CRIT."
1080COLOUR1
1090COLOUR128
1100PRINTTAB(24,12);
1110IFDX>0THENPRINT"SHIELD FAILURE":SOUND1,-15,3
0,5ELSEPRINT"
"
1120PRINTTAB(16,12);RIGHT$("      "+STR$(DTZDIV2)
,5)
1130RETURN
1135
1140REM===== COMMAND ROUTINE
1145
1150PRINTTAB(3,15)"COMMAND ";
1160C$=INKEY$(0)
1170T1=TIME+99
1180PRINTTAB(3,15)"COMMAND *";
1190IFC$<>" "THENSOUND1,-13,160,1ELSE1280
1200IFC$="F"THEN1970
1210IFC$="M"THEN2480
1220IFC$="L"THEN3260
1230IFC$="S"THEN3510
1240IFC$="E"THEN3020
1250IFC$="R"THEN4210
1260IFC$="P"THENSOUND1,-15,120,5:SOUND1,-12,20,5
:60T02450
1270REM Update klingons etc. & wait for the rem
ainder of the second.
1280GOSUB1320
1290REPEATUNTILTIME>T1
1300GOTO1150
1305
1310REM===== MOVE KLINGONS , COUNT TIME
1315
1320IFFL>0THENFL=FL-SR ELSE3190

```

```

1330HT=KQZ*5*RND(1)
1340DZ=HT-SR
1350IFDZ>0THENIX=RND(4)-1:DZ(IX)=DZ(IX)-DZ:DTZ=D
TZ+DZ:IFDTZ>200THEN3230
1360MXZ=RND(10)-1
1370MYZ=RND(10)-1
1380IFMXZ=IXZANDMYZ=IYZTHEN1450
1390IFGZ(MXZ,MYZ)<9900THEN1450
1400GZ(MXZ,MYZ)=GZ(MXZ,MYZ)-10000
1410M1Z=SGN(IXZ-MXZ)*RND(ABS(IXZ-MXZ))
1420M2Z=SGN(IYZ-MYZ)*RND(ABS(IYZ-MYZ))
1430GZ(MXZ+M1Z,MYZ+M2Z)=GZ(MXZ+M1Z,MYZ+M2Z)+1000
0
1440IFMXZ+M1Z=IXZANDMYZ+M2Z=IYZTHENLLZ=1:NZ=1000
0:GOSUB1890:PROCD(AK,QXZ,QYZ):KQZ=KQZ+1
1450IFQZ(MXZ,MYZ)<>10000THEN1500 ELSEQZ(MXZ,MYZ)
=0:PROCD(32,MXZ,MYZ)
1460LLZ=1
1470NZ=10000
1480GOSUB1890
1490PROCD(AK,QXZ,QYZ)
1500GOSUB980
1510IFDZ(2)<0THENSZ=0
1520RETURN
1530
1540REM Init damage levels to undamaged
1545
1550DZ(0)=10
1560DZ(1)=15
1570DZ(2)=20
1580DZ(3)=10
1590DTZ=0
1600RETURN
1605
1610REM= SET UP QUADRANT ROUTINE
1615
1620NXZ=(XE-INT(XE))*10
1630NYZ=(YE-INT(YE))*10
1640FORLAZ=0TO9
1650FORLBZ=0TO9
1660QZ(LAZ,LBZ)=0
1670NEXT

```

```

1680NEXT
1690G=GZ(INT(XE),INT(YE))
1700QZ(NXZ,NYZ)=-10
1710IXZ=XE
1720IYZ=YE
1730ZZ=G
1740GOSUB1850
1750NZ=1
1760GOSUB1890
1770GOSUB1850
1780NZ=100
1790GOSUB1890
1800GOSUB1850
1810NZ=10000
1820KQZ=LLZ
1830GOSUB1890
1840RETURN
1850H=ZZ/100
1860LLZ=(H-INT(H))*100+.5
1870ZZ=INT(H+.5)
1880RETURN
1890IFLLZ<=0THENRETURN ELSEFORLZ=1TOLLZ
1900QXZ=RND(10)-1
1910QYZ=RND(10)-1
1920IFQZ(QXZ,QYZ)<>0THEN1900
1930QZ(QXZ,QYZ)=NZ
1940NEXT
1950RETURN
1955
1960REM=== FIRING ROUTINE
1965
1970PROCCON
1980IFPT<=0THENPRINT"ALL PHOTON TORPEDOES FIRED"
:PROCCUPD(3,TRUE):PROCCOFF:GOTO1150
1990IFDZ(0)<0THENPRINT"TORPEDO BANKS DAMAGED":PR
OCUPD(3,TRUE):PROCCOFF:GOTO1150
2000PRINT"TORPEDO BANKS";11-PT;" TO 10 READY TO
FIRE"
2010PRINT"DIRECTION (S=TO STOP) ";
2020TS=TIME
2030INPUTLINE"====>>>"C$

```

```

2040IFC$="S"THENPROCUPD((TIME-TS)DIV100,FALSE):P
ROCCOFF:GOTO1150
2050PT=PT-1
2060PROCUPD((TIME-TS)DIV100,FALSE)
2070PROCCOFF
2080V=VALC$
2090XF=SIN(V/180*PI)
2100YF=COS(V/180*PI)
2110TV=.14
2120SOUND1,-10,240,5
2130GOSUB1320
2140TV=TV+.95
2150X5=NXZ+TV*XF
2160Y5=NYZ+TV*YF
2170X6=X5+.1
2180Y6=Y5+.1
2190IFX6<0ORX6>9.999ORY6<0ORY6>9.999THENPROCON:P
RINT"BANK ";10-PT;" FIRED":PROCCOFF:GOTO2420
2200IFINT(X6)=NXZANDINT(Y6)=NYZTHEN2130
2210IFQZ(X6,Y6)=0THENPROCD(PHO,X6,Y6):SOUND0,-10
+TV,5+(TV AND1),1:PROCD(32,X6,Y6)
2220IFQZ(X6,Y6)<=0THEN2130 ELSEPROCON
2230IFQZ(X6,Y6)=100THENPRINT" * *YOU NIT YOU SHO
T A STARBASE * *":PROCCOFF:GOTO2290
2240IFQZ(X6,Y6)<>1THEN2270
2250PRINT"TORPEDO HIT A STAR"
2260IFRND(9)<7THENPROCCOFF:GOSUB4130:F0RZS=0T025
:SOUND0,-15,ZS AND7,1:NEXT:GOTO3610 ELSE2420
2270IFQZ(X6,Y6)=10000THENKQZ=KQZ-1:SOUND0,-15,6,
3
2280PROCCOFF
2290FORJK=0T010
2300SOUND0,-15,7,10
2310SOUND1,-1,JK*10,10
2320NEXT
2330QZ(X6,Y6)=0
2340QX=X6
2350QY=Y6
2360FORZS=0T0100
2370PROCD(EX,X6,Y6)
2380PROCD(42,X6,Y6)
2390NEXT

```

```

2400PROCD(32,X6,Y6)
2410GOTO1970
2420PROCON
2430PROCUPD(4,TRUE)
2440GOTO1970
2450IFDZ(1)>0THENC1$=STR$RND(360):C2$="3":GOTO2
580
2460
2470REM ===Warp drive routine
2475
2480PROCCON
2490IFDZ(1)<0THENPRINT"ENGINES DAMAGED":PROCCOFF:
REPEATGOSUB1320:UNTILINKEY(3)<>TRUE:PROCON:PROCCO
FF:GOTO1150
2500TS=TIME
2510INPUTLINE"GIVE HEADING ++--->",C1$
2520PROCUPD((TIME-TS)DIV100,FALSE)
2530TS=TIME
2540INPUTLINE"GIVE DISTANCE++--->",C2$
2550PROCUPD((TIME-TS)DIV100,FALSE)
2560IFINSTR(C1$+C2$+" ","S")<>0THENPROCCOFF:GOTO
1150
2570PROCCOFF
2580QZ(NXZ,NYZ)=0
2590G=0
2600FORIX=0TO9
2610FORJZ=0TO9
2620G=G+QZ(IX,JZ)
2630NEXT
2640NEXT
2650QZ(NXZ,NYZ)=-10
2660GZ(IXZ,IYZ)=G
2670HA=VALC1$
2680HD=VALC2$
2690FL=FL-ABS(HD*50)
2700XE=XE+SIN(HA*PI/180)*HD
2710YE=YE+COS(HA*PI/180)*HD
2720KXZ=XE
2730KYZ=YE
2740QZ(NXZ,NYZ)=0
2750SOUND1,1,10,40

```

```

2760IFNOT(KXZ=IXZANDKYZ=IYZ)THEN2950
2770PROCD(32,NXZ,NYZ)
2780NXZ=(XE-IXZ)*10
2790NYZ=(YE-IYZ)*10
2800IFQZ(NXZ,NYZ)=100THENGOSUB1550:PROCD(E,NXZ,N
YZ):QZ(NXZ,NYZ)=-10:PT=10:FL=FL+7000:GOTO3860
2810IFQZ(NXZ,NYZ)<>0THEN4170
2820QZ(NXZ,NYZ)=-10
2830PROCD(E,NXZ,NYZ)
2840GOTO1150
2850PROCCON
2860PRINT"YOU LEFT THE GALAXY. RANDOM " "RE-ENTR
Y ACTIVATED"
2870FORJK=1TO5
2880FORJL=1TO100
2890NEXT
2900SOUND1,-15,120,2
2910SOUND1,-15,70,2
2920NEXT
2930XE=RND(9)-RND(0)
2940YE=RND(9)-RND(0)
2950PROCCON
2960PROCCOFF
2970IXZ=INTXE
2980IYZ=INTYE
2990IFNOT(IXZ>=0ANDIYZ>=0ANDIXZ<=9ANDIYZ<=9)THEN
2850
3000GOSUB1620
3010GOTO820
3020PROCCON
3030G=0
3040Z=0
3050QZ(NXZ,NYZ)=0
3060GZ(IXZ,IYZ)=0
3070FORIX=0TO9
3080FORJZ=0TO9
3090G=G+GZ(IX,JZ)
3100Z=Z+QZ(IX,JZ)
3110NEXT
3120NEXT
3130G=G+Z
3140IFG>=10000THENPRINTINT(G/10000);"KLINGONS LE

```

```

FT":PROCCOFF:FORI1Z=0TO10:GOSUB1320:NEXT:PROCON:PR
OCCOFF:GOTO1150
3150PRINT"/"/"**** WELL DONE ****"
3160*FX15,1
3170A=GET
3180RUN
3190FL=0
3200GOSUB980
3210PRINTTAB(0,14)" OUT OF FUEL LIFE SUPPORT FA
ILED "/TAB(11)"YOU'RE DEAD !!!!!!!!!!...";
3220GOTO3160
3230GOSUB980
3240PRINTTAB(10,14)" ENTERPRISE DESTROYED ";
3250GOTO3160
3260PROCCON
3270IFDZ(3)<0THENPRINT"SCANNER DAMAGED ":PROCUPD(
3,TRUE):PROCCOFF:GOTO1150 ELSEPRINTTAB(4)" S B
K S B K S B K"
3280QZ(NXZ,NYZ)=0
3290G=0
3300FORIUZ=0TO9
3310FORICZ=0TO9
3320G=G+QZ(IUZ,ICZ)
3330NEXT
3340NEXT
3350GZ(IXZ,IYZ)=G
3360QZ(NXZ,NYZ)=-10
3370PROCUPD(1,FALSE)
3380C$=INKEY$(0)
3390IFC$<>" "THENPROCCOFF:GOTO1180
3400FORIUZ=(IXZ>0)TO-(IXZ<9)
3410FORICZ=(IYZ>0)TO-(IYZ<9)
3420G$=STR$(GZ(IXZ+IUZ,IYZ+ICZ))
3430LZ$=LENG$
3440G$=STRING$(6-LZ$,"0")+G$
3450PRINTTAB(5+11*(IUZ+1),4+2*(ICZ+1));MID$(G$,5
,2);" ";MID$(G$,3,2);" ";MID$(G$,1,2)
3460NEXT
3470NEXT
3480C$=INKEY$(60)
3490IFC$<>" "THENPROCCOFF:GOTO1180

```



```

3500IFDZ(3)<0THENPROCCOFF:GOTO3260ELSE3280
3510PROCCON
3520PRINT"NEW SHIELD POWER";
3530TS=TIME
3540INPUTLINEC$
3550PROCCOFF
3560FORI=1TO(TIME-TS)DIV100
3570GOSUB1320
3580NEXT
3590SR=VALC$
3600GOTO1150
3610FORIZ=1TO900STEP50
3620SOUND0,-15,5,2
3630
3640FORJZ=1ZTO1000
3650NEXT
3660VDU19;7;0;
3670FORJZ=1ZTO100
3680NEXT
3690VDU19;0;0;
3700NEXT
3710MODE5
3720FORI=0TO50
3730GCOL0,RND(3)
3740SE=RND(1)*2*PI
3750MOVE640,512
3760DRAW640+I*10*SIN(SE),512+I*10*COS(SE)
3770SOUND0,-15+I/8,(I OR4)AND7,2
3780NEXT
3790MODE4
3800FORIZ=0TO7000
3810NEXT
3820GZ(IXZ,IYZ)=0
3830FL=FL-5000
3840SR=RND(SR)
3850GOTO590
3860CLS
3870PRINT""
3880PRINT"      ENTERPRISE DOCKED AT STARBASE"
3890PRINT"      FUEL BEING LOADED"
3900PRINT"      TORPEDO BANKS BEING REARMED"
3910PRINT"      REPAIRS COMPLETE"

```

```

3920FORI=0T09000
3930NEXT
3940MXZ=RND(10)-1
3950MYZ=RND(10)-1
3960GZ(MXZ,MYZ)=GZ(MXZ,MYZ)+100
3970CLS
3980PRINTTAB(7,29)"STARBASE LEAVING";
3990U=159
4000V=159
4010VDU5
4020GCOL3,1
4030REPEAT
4040MOVEU,V
4050VDUAB
4060MOVEU,V
4070U=U+U/150
4080V=V+V/200
4090VDUAB
4100UNTILV>1024
4110VDU4
4120GOTO660
4130PROCOFF
4140PROCD(SUP,X6,Y6)
4150PROCON
4160RETURN
4170CLS
4180PRINT///// "BANG ! ! ! ! ! ! ! !....."
4190PRINT///// "I THINK YOU HIT SOMETHING BIG"///
/"PS.  YOU'RE VERY DEAD',',',',',....."
4200GOTO3160
4210PROCCON
4220PRINT"REPAIRS STARTED";CHR$26
4230PROCUPD(DTZDIV5,TRUE)
4240FL=FL-5*DTZ
4250GOSUB1550
4260PRINT" THEY ARE NOW COMPLETED"
4270PROCUPD(3,TRUE)
4280PROCCOFF
4290GOTO1150
4300
4310REM {OS 1.0 ONWARDS ONLY}Debugging routine n

```

```

ot important just lists error line.
43200NERROROFF
4330REPORT
4340IFERL<>0THENPRINT" at line ";ERL
4350E$="L."+STR$ERL+CHR$13
4360XZ=0
4370AX=138
4380*FX15,1
4390FORIX=1TOLENE$
4400YZ=ASCID$(E$,IX,1)
4410CALL&FFF4
4420NEXT
4430END
4440
4450DEFPROCDC(C,X,Y) PRINTTAB(X+1,Y+1);
4460VDUC,30
4470ENDPROC
4475
4480REM Display object on short range scanner
4485
4490
4500DEFPROCCON
4510VDU28,1,30,38,15
4520CLS
4530PRINT" COMMAND :";C$
4540ENDPROC
4545
4550
4560DEFPROCCOFF
4570COLOUR129
4580CLS
4590COLOUR128
4600VDU26
4610ENDPROC
4615
4620
4630DEFPROCON VDU28,1,30,38,15
4635
4640REM Put text window around commandarea
4645
4650PRINTTAB(HP,VP)

```

```

4660ENDPROC
4665
4670
4680DEFPROC OFF
4685
4690REM Expand text command screen to full screen
n
4695
4700HP=POS
4710VP=VPOS
4720VDU26
4730ENDPROC
4735
4740
4750DEFPROC UPD(N,SLOW) LOCAL IZ,TS
4755
4760REM Update kilings etc. & If SLOW true wait
it for the remainder of the second.
4765
4770PROC OFF
4780FOR IZ=1 TO N
4790TS=TIME+99
4800GOSUB 1320
4810IF SLOW THEN REPEAT UNTIL TIME>TS
4820NEXT IZ
4830PROC ON
4840ENDPROC

```

SPACE-BLASTER

This is a 3-D space war game. You are in a small Star Fighter class space ship (your ship is displayed as part of the title display), the aliens are using saucers of an unknown type. Your orders are to blast them but the aliens are not too keen on this idea and shoot back. The controls for your ship are the keys "A", "Z", "<", ">", & the space bar. In this battle you can survive your ship being blown up three times but after this you are very dead. The computer will then give you a comment on your score.

MAIN ROUTINES

Main Program: Lines 50-120 puts screen in mode two, displays intro, initialises arrays etc. and loops for each time your ship gets blown up.

PROCLIFE: Fight aliens until your ship destroyed.

PROCINIT: Dimension arrays, position stars, define chars & initialise main variables.

PROCINITGO: Initialise alien positions.

PROCSF: Move stars.

PROCMOVEE: Move aliens.

PROCOBK: Check keys and do movement and firing accordingly.

PROCALIEN: Draw correct size alien on screen.

PROCUP: Move stars & aliens down.

PROCDOWN: Move stars & aliens up.

PROCRIGHT: Move stars etc left.

PROCLEFT: Move stars etc right.

PROCSIGHT: Draw gun sight and display score.

PROCFIRE: Fire laser check for hits & do score.

PROCHIT: Alien fire hit you.

PROCMISS: Alien fire missed you.

PROCSCORE: Display score.

PROCCOMMENT: Comments on player's score at end of game.

PROCINTRO: Draw initial display, show title and give instructions.

```

10REM*****
20REM*****SPACE-BLASTER*****
30REM*****
40
50MODE2
60PROCINTRO
70PROCINIT
80REPEAT
90PROCLIFE
100UNTILLIFE=0
110PROCCOMMENT
120END
130
140
150DEFPROCLIFE
160DEAD=FALSE
170PROCINITGO
180PROCSIGHT
190REPEAT
200PROCSF
210PROCMOVEE
220PROCGBK
230UNTILDEAD
240LIFE=LIFE-1
250ENDPROC
260
270
280DEFPROCINIT
290
300REM Number of aliens
310
320NE=3
330
340REM Number of stars
350
360NS=5
370
380REM Star field,Enemy position,Enemy velocity
& Star position arrays
390
400DIM SFZ(NS,1),EPZ(NE,2),EVZ(NE,2),SPZ(NS,1)

```

```

410FORIX=0TONS
420SFZ(IX,0)=RND(1280)-640
430SFZ(IX,1)=RND(1024)-512
440NEXT
450SPZ=0
460LIFE=3
470SCOREZ=0
480
490REM Define characters for enemy sauser at 4
distances
500
510VDU23,224,0,0,7,14,31,2,4,4
520VDU23,225,60,126,255,170,255,126,24,24
530VDU23,226,0,0,224,176,248,64,32,32
540VDU23,227,0,0,24,60,255,60,0,0
550VDU23,228,0,0,0,16,56,0,0,0
560VDU23,229,0,0,0,16,0,0,0,0
570ENVELOPE1,1,1,1,1,0,0,0,120,-3,0,-1,126,80
580ENVELOPE2,1,50,-50,-3,4,1,1,120,-3,-1,-1,126
,80
590ENDPROC
600
610
620DEFPROCINIT60
630
640REM Make them start by attacking from behind
650
660FORIX=0TONE
670EPZ(IX,0)=RND(2000)-1000
680EPZ(IX,1)=RND(2000)-1000
690EPZ(IX,2)=RND(100)-120
700NEXT
710FORIX=0TONS
720SPZ(IX,0)=0
730SPZ(IX,1)=0
740NEXT
750ENDPROC
760
770
780DEFPROCASF
790
800REM Move stars

```

```

810
820FORIX=0TONS
830GCOL3,7
840PLOT69,SPZ(IX,0),SPZ(IX,1)
850SPZ=SPZ+24
860SPZ(IX,0)=SFZ(IX,0)*((SPZ+IX)AND127)DIV127
870SPZ(IX,1)=SFZ(IX,1)*((SPZ+IX)AND127)DIV127
880PLOT69,SPZ(IX,0),SPZ(IX,1)
890NEXT
900ENDPROC
910
920
930DEFPROCMOVEE
940
950REM Move aliens
960
970FORIX=0TONE
980PROCALIEN(IX)
990FORJZ=0T02
1000EPZ(IX,JZ)=EPZ(IX,JZ)+EVZ(IX,JZ)
1010NEXT
1020PROCALIEN(IX)
1030NEXT
1040NZ=RND(10)-1
1050IFNZ>3THENENDPROC
1060EVZ(NZ,0)=-SGN(EPZ(NZ,0))*RND(12)
1070EVZ(NZ,1)=-SGN(EPZ(NZ,1))*RND(12)
1080EVZ(NZ,2)=-SGN(EPZ(NZ,2)-70)*RND(5)
1090ZZ=EPZ(NZ,2)
1100IFZZ<200RZZ>70THENENDPROC
1110IFRND(2)=2THENENDPROC
1120XZ=EPZ(NZ,0)
1130YZ=EPZ(NZ,1)
1140IFXZDIV100DRYZDIV100THENPROCHMISS ELSEPROCHIT
1150ENDPROC
1160
1170
1180DEFPROC0BK
1190
1200REM Obey key input
1210

```



```

1220IFINKEY(-66)THENPROCUP
1230IFINKEY(-98)THENPROCDOWN
1240IFINKEY(-103)THENPROCLEFT
1250IFINKEY(-104)THENPROCRIGHT
1260IFINKEY(-99)THENPROCFIRE
1270ENDPROC
1280
1290
1300DEFPROCALIEN(NZ)
1310
1320REM Display alien on screen
1330
1340DX=EPZ(NZ,2)
1350IFDX<20THENENDPROC
1360PE=50/DX
1370MOVEEPZ(NZ,0)*PE-96,EPZ(NZ,1)*PE+16
1380GCOL3,NZ+1
1390IFDX>100THENVDU5,32,229,32,4:ENDPROC
1400IFDX>60THENVDU5,32,228,32,4:ENDPROC
1410IFDX>40THENVDU5,32,227,32,4:ENDPROC
1420VDU5,224,225,226,4
1430ENDPROC
1440
1450
1460DEFPROCUP
1470FORIX=0TONS
1480SFZ(IX,1)=(SFZ(IX,1)+572AND1023)-512
1490NEXT
1500FORIX=0TONE
1510PROCALIEN(IX)
1520EPZ(IX,1)=EPZ(IX,1)-60
1530PROCALIEN(IX)
1540NEXT
1550ENDPROC
1560
1570
1580DEFPROCDOWN
1590FORIX=0TONS
1600SFZ(IX,1)=(SFZ(IX,1)+1576AND1023)-512
1610NEXT
1620FORIX=0TONE

```

```

1630PROCALIEN(IZ)
1640EPZ(IZ,1)=EPZ(IZ,1)+60
1650PROCALIEN(IZ)
1660NEXT
1670ENDPROC
1680
1690
1700DEFPROCRIGHT
1710FORIZ=0TONS
1720SFZ(IZ,0)=((SFZ(IZ,0)+1860)MOD1280)-640
1730NEXT
1740FORIZ=0TONE
1750PROCALIEN(IZ)
1760EPZ(IZ,0)=EPZ(IZ,0)-60
1770PROCALIEN(IZ)
1780NEXT
1790ENDPROC
1800
1810
1820DEFPROCLEFT
1830FORIZ=0TONS
1840SFZ(IZ,0)=((SFZ(IZ,0)+700)MOD1280)-640
1850NEXT
1860FORIZ=0TONE
1870PROCALIEN(IZ)
1880EPZ(IZ,0)=EPZ(IZ,0)+60
1890PROCALIEN(IZ)
1900NEXT
1910ENDPROC
1920
1930
1940DEFPROCSTIGHT
1950
1960REM Move graphics origin to centre of screen
& draw gun sight
1970
1980CLS
1990VDU29,640;512;
2000MOVE-64,0
2010GCOLOR,14
2020PLOT21,64,0

```

```

2030MOVE0,-64
2040PLOT21,0,64
2050PROCSCORE
2060ENDPROC
2070
2080
2090DEFPROCFIRE LOCAL XZ,YZ,ZZ
2100
2110REM Fire guns & check for a hit
2120
2130PROCZAP
2140FORIX=0TONE
2150ZZ=EPZ(IX,2)
2160IFZZ<200RZZ>70THENNEXT:ENDPROC
2170XZ=EPZ(IX,0)DIV58
2180YZ=EPZ(IX,1)DIV16
2190IFXZORYZTHENNEXT:ENDPROC
2200PROCSCORE
2210SCOREZ=SCOREZ+10
2220PROCSCORE
2230SOUND0,1,5,5
2240PROCALIEN(IX)
2250EPZ(IX,2)=-100
2260IX=NE
2270NEXT
2280ENDPROC
2290
2300
2310DEFPROCZAP
2320
2330REM Graphics & sound effects for your gun fi
re
2340
2350SOUND1,2,100,15
2360FORGZ=0T01
2370FORCZ=1T08
2380GCOL3,CZ
2390MOVE-600,-512
2400DRAW0,0
2410MOVE600,-512
2420DRAW0,0

```

```

2430NEXT
2440NEXT
2450ENDPROC
2460
2470
2480DEFPROCHIT
2490
2500REM You are being Zapped
2510
2520SOUND0,1,4,30
2530PE=50/EPZ(NZ,2)
2540XZ=XZ*PE
2550YZ=YZ*PE
2560FORJZ=0TO70
2570GCOL3,JZMOD6+1
2580MOVEXZ,YZ
2590X1Z=RND(1280)-640
2600Y1Z=RND(1024)-512
2610DRAWX1Z,Y1Z
2620NEXT
2630DEAD=TRUE
2640ENDPROC
2650
2660
2670DEFPROCHMISS
2680
2690REM He missed you
2700
2710PE=50/EPZ(NZ,2)
2720XZ=XZ*PE
2730YZ=YZ*PE
2740FORJZ=0TO8
2750GCOL3,JZMOD4+1
2760MOVEXZ,YZ
2770X1Z=SGNXZ*700+RND(50)-25
2780Y1Z=SGNYZ*512+RND(50)-25
2790DRAWX1Z,Y1Z
2800DRAWXZ,YZ
2810NEXT
2820ENDPROC
2830

```

```

2840
2850DEFPROCSCORE
2860
2870REM Display or delete score
2880
2890GCOL3,6
2900MOVE-640,508
2910PRINTCHR$5"SCORE:";SCOREZ"           "LIFE CHR$
4;
2920ENDPROC
2930
2940
2950DEFPROCCOMMENT
2960CLS
2970PRINTTAB(0,5)"You scored ";SCOREZ
2980FORIX=0TOSCOREZ/100
2990READA$
3000NEXT
3010IFSCOREZ>850THENPRINT"SHOW OFF":ENDPROC
3020PRINTTAB(0,9)"THAT IS "A$"."
3030*FX15,1
3040ENDPROC
3050
3060
3070DEFPROCINTRO
3080
3090REM Text window at bottom of the screen
3100
3110VDU28,0,31,19,20
3120
3130REM Plot some stars
3140
3150GCOLOR,7
3160FORIX=0TO30
3170PLOT69,RND(1279),RND(700)+323
3180NEXT
3190
3200REM Draw your star fighter
3210
3220FORI=-.4TO.4STEP.8
3230GCOLOR,5

```

```

3240MOVE1000,700
3250MOVE400,700+300*I
3260PLOT85,350,700+100*I
3270GCOLOR,6
3280MOVE490,700+200*I
3290MOVE410,700+450*I
3300PLOT85,400,700+210*I
3310PLOT85,370,700+450*I
3320NEXT
3330GCOLOR,4
3340MOVE1000,700
3350MOVE310,744
3360PLOT85,310,654
3370GCOLOR,7
3380MOVE800,740
3390DRAW750,780
3400DRAW710,780
3410DRAW590,785
3420PLOT85,710,760
3430MOVE306,720
3440GCOLOR,1
3450DRAW306,680
3460
3470REM Print up title
3480
3490DIMT 14
3500$T="SPACE-BLASTER"
3510FORJ=96TO100
3520GCOLOR,(J+4) DIV4MOD7+1
3530FORI=-6TO6
3540MOVE624+I*J,J*2
3550VDUS,(T+I+6),4
3560NEXT
3570NEXT
3580COLOUR3
3590TIME=0
3600REPEATUNTILTIME>300
3610PRINT"/////////" SPACE BAR TO BLAST"/" THEM
"
3620PRINT""A','Z','<' & '>'"" TO CONTROL SHIP
"

```

3630TIME=0
3640REPEATUNTILTIME>400
3650VDU26
3660ENDPROC
3670DATA TERRIBLE,NOT VERY GOOD,NOT BAD,REASONAB
LE,OK,GOOD,GREAT,FUNNY. DID YOU CHEAT

BRICK BASHER

This is a breakout type game. The object of the game is to break through as many advancing walls of bricks as possible, by bouncing a ball (antimatter particle in disguise) against bricks (made of lead?) using your bat (moveable magnetic mirror). The game has more than one kind of brick for you to bash, each type of brick behaves differently and gives a different score when hit. Your bat is moved sideways using "<" & ">" keys. When you lose a ball off the bottom of the screen a new ball is served after a short pause until all the allowed balls have been used.

When the game is RUN it first asks for a volume for sound effects 0 = no sound, 9 = max. volume. The title and instructions are then printed and the game starts when any key is pressed. At the end of each game it gives you the option to play again.

MAIN PROCEDURES & FUNCTIONS

Main Prog: Lines 10-830 do the title and initialisation. Lines 840-1110 Main program loop.

PROCWALL: Draws up walls made of different type of bricks the type of wall depends on the variable LEV.

PROCW: Draw a wall of brick type C.

PROCINIT: Sets up the initial position and direction of movement of a new ball.

PROCMB: Test "<" & ">" keys and move bat accordingly.

PROCH: Move ball horizontally and check for collisions with bricks, bat and sides of play area.

PROCV: Move ball vertically and check for collisions.

PROCSM: Destroys brick and updates the score.

PROCR: Places N bricks of type C in random positions in a new wall.

PROCUP: Move bricks up again N lines used when you hit a certain type of brick.


```

2MODE7
5PRINT"/"VOLUME (0-9):";
6VD=VALGET$/9
10MODE2
20ENVELOPE1,129,7,2,-7,1,1,2,124,-2,0,-1,126*V
0,30*VD
30ENVELOPE2,129,1,1,-1,3,1,2,124,-2,0,-1,100*V
0,40*VD
40ENVELOPE3,1,1,-1,1,1,1,1,60,-2,0,-1,126*VD,1
20*VD
50ENVELOPE4,1,0,-1,0,1,1,1,50,-1,0,-1,100*VD,5
0*VD
55ENVELOPE5,1,-12,+12,-12,1,1,1,100,-1,0,-20,1
00,20
56
57REM Redefine colours 15,11,12
58
60VDU19,15,4;0;19,11,6;0;19,12,0;0;
64
65REM Define brick character
66
70BR=224
80VDU23,BR,255,255,255,195,195,195,255,255
84
85REM Define ball character
86
90BAZ=226
100VDU23,BAZ,192,192,0,0,0,0,0,0
110W$=STRING$(18,CHR$(BR))
114
115REM Define bat characters
116
120B1Z=227
130VDU23,B1Z,0,63,255,255,255,255,255,255
140B2Z=228
150VDU23,B2Z,255,255,255,255,255,255,255,255
160B3Z=229
170VDU23,B3Z,0,252,255,255,255,255,255,255
174
175REM Set up strings to pront multi-coloured b
at
176

```

```

180B0$=CHR$(18)+CHR$(0)+CHR$(0)+CHR$(B1Z)+CHR$(
B2Z)+CHR$(B3Z)
190BC$=CHR$(18)+CHR$(0)+CHR$(11)+CHR$(B1Z)+CHR$
(18)+CHR$(0)+CHR$(15)+CHR$(B2Z)+CHR$(18)+CHR$(0)+
CHR$(11)+CHR$(B3Z)
200COLOUR143
201CLS
210COLOUR2
211PRINTTAB(0,31);"Score";
220VDU28,1,30,18,1
230VDU30
240COLOUR14
250PRINT"    BRICK BASHER"
260COLOUR5
270PRINT"/""<""&"">""TO MOVE BAT"
280*FX15,1
290PRINT"/""PRESS ANY KEY"/""TO PLAY:"
300A$=GET$
301COLOUR128
310LZ=65
311RZ=980
312K1Z=-103
313K2Z=-104
320YMZ=60
321SZ=32
322ZZ=SZ*2
323SCZ=0
324DZ=10
330PROCW(10)
334
335REM Scroll intro of screen
336
340FORI=1TO26
341PROCW(0)
342NEXT
350LEV=0
351BATS=3
352NB=0
353BZ=600
354
355REM Draw up starting walls

```

```

356
360FORI=0T04
361PROCW(1)
362PROCR(2,7)
363NEXT
370PROCW(0)
380
385REM Loop for each new ball
386
390REPEAT
400PROCINIT
410
415REM Alternatly move ball horizontaly & verti
cally
416
420REPEATCZ=CZEORTRUE
430IFCZTHENPROCV:PROCVB ELSEPROCH
440UNTILYZ<YHZ
444
445REM Ball of bottom of the play area
450
460SOUND2,3,40,20
470MOVEXZ,YZ
471GCOL0,0
472VDU5,BAZ
473BATS=BATS-1
480PROCDEL(200)
490UNTILBATS=-1
500VDU26,4
501FORI=0T036
502VDU10
503PROCDEL(20)
504NEXT
510COLOUR7
511PRINT'/'/"PRESS ANY KEY : "
520*FX15,1
530A$=GET$
540RUN
550
555
556

```

```

560DEFPROCWALL
570LOCALL,C
580IFLEV<3THENPROCW(1):ENDPROC
590IFLEV<5THENPROCW(0):ENDPROC
600IFLEV<9THENPROCW(2):PROCR(1,7):ENDPROC
610IFLEV<10THENPROCW(0):ENDPROC
620IFLEV<13THENPROCW(3):PROCR(4,15):ENDPROC
630IFLEV<16THENPROCW(0):ENDPROC
640IFLEV<20THENPROCW(4):PROCR(3,15):ENDPROC
650IFLEV<22THENPROCW(0):ENDPROC
655IFLEV<25THENPROCW(5):PROCR(1,12):ENDPROC
660LEV=0
661DX=DX-2
670ENDPROC
675
680
690DEFPROCW(C)
700VDU4,30,11,30
710$UND0,4,6,3-
720COLOURC
730PRINTW$;
740VDU5
750ENDPROC
755
760
770DEFPROCINIT
780XZ=640
781YZ=180
790REPEATDXZ=(RND(3)-2)*SZ
791UNTILDXZ
800DYZ=SZ
801CZ=FALSE
810MOVEXZ,YZ
811GCOL3,7
812VDUBAZ
820GCOL0,0
821MOVEBZ,64
830BZ=576
831VDUB1Z,B2Z,B3Z
840ENDPROC
845

```

```

850
860DEFPROCMB
865LOCALM%
866
867REM Move bat
868
870IFINKEY(K1%)ANDB%>L%THENM%=-Z%
880IFINKEY(K2%)ANDB%<R%THENM%=Z%
890MOVEB%,64
891PRINTB0$;
892B%=B%+M%
895MOVEB%,64
896PRINTB0$;
897ENDPROC
898
900
910DEFPROCDEL(L)
920L=TIME+L
930REPEATUNTILTIME>L
940ENDPROC
945
950
960DEFPROCV
964
965REM Move ball vertically
966
970X1%=X%
971Y1%=Y%+DY%
972MOVEX%,Y%
973GCOL0,0
974VDUBAZ
980V%=POINT(X1%,Y1%)
990IFV%=0THENX%=X1%:Y%=Y1%:MOVEX%,Y%:GCOL0,7:VD
UBAZ:ENDPROC
1000NB=NB+1
1010IFV%=15THENDY%=-DY%:SOUND3,2,160,5:GOTO1200
1020IFV%<>11THENDY%=-DY%:SOUND3,1,180,6:PROCSM:G
OTO1200
1030DY%=SZ
1040IFX%<B%+72THENDX%=-SZELSEDX%=SZ
1050SOUND1,2,150,7

```

```

1060GOTO1200
1070
1080DEFPROC H
1084
1085REM Move ball horizontally
1086
1090X1Z=XZ+DXZ
1091Y1Z=YZ
1092MOVEXZ,YZ
1093GCOL0,0
1094VDUBAZ
1096
1100VZ=POINT(X1Z,Y1Z)
1110IFVZ=0THENXZ=X1Z:YZ=Y1Z:MOVEXZ,YZ:GCOL0,7:VD
UBAZ:ENDPROC
1120NB=NB+1
1130IFVZ=15THENDXZ=-DXZ:SOUND1,2,160,5:GOTO1200
1140IFVZ<>11THENDXZ=-DXZ:SOUND1,1,180,6:PROCSM:G
OTO1200
1150DYZ=SZ
1160DXZ=-DXZ
1170SOUND1,2,150,7
1180GOTO1200
1190
1200IFNB>DXTHENPROCWALL:LEV=LEV+1:NB=0
1210MOVEXZ,YZ
1211GCOL0,7
1212VDUBAZ
1220ENDPROC
1230
1240DEFPROC SM
1250LOCALXZ,YZ
1254
1256REM Destroy brick
1257
1260XZ=X1ZDIV64
1270YZ=31-Y1ZDIV32
1280VDU26,4,30
1281PRINTTAB(XZ,YZ);" ";
1285IFVZ=12THENCOLOUR141:PRINTTAB(XZ,YZ);" ";:SO
UND2,4,200,5:SCZ=SCZ+100:COLOUR128

```

```

1290SCZ=SCZ+VZ
1300COLOUR7
1310PRINTTAB(7,31);SCZ;
1315PRINTTAB(15,31);BATS;
1320VDU28,1,30,18,1,5
1325IFVZ=7THENSOUND0,4,5,2:PROCW(0)
1327IFVZ=13THENPROCUP(3)
1330ENDPROC
1340
1345
1350DEFPROC(R,N,C)
1360LOCALIZ
1361
1362REM Change some of the bricks in new wall
1363
1365COLOURC
1370FORIX=1TON
1380VDU4
1381PRINTTAB(RND(17),0);CHR$(BR);
1390NEXT
1391VDU5
1392ENDPROC
1400
1405
1410DEFPROCUP(N)
1414
1415REM Scroll screen back
1416
1420VDU26,4,28,1,29,18,1
1430PRINTSTRING$(28+N,CHR$(10));
1440VDU28,1,30,18,1,5
1450SOUND1,1,120,7
1460SOUND2,1,128,7
1470SOUND3,1,128,7
1480ENDPROC

```

SPACE INVADERS

This is space invaders written in basic so do not expect fantastic speed from the advancing aliens but to make up for this the aliens drop unpowered target seeking missiles instead of bombs. If you kill all of the first wave of aliens then another wave will appear but starting lower than the first. This is repeated until you have lost all of your bases or the aliens land at the bottom of the screen.

When the game is RUN it prints up the title and displays instructions for using the "<", ">" & "F" keys. The game is then started by pressing the space bar.

MAIN ROUTINES

Main Program: Line 10-880 initialise characters, envelopes, arrays and main variables. The loop for moving objects around the screen starts at 600. Each time around the loop one alien is moved, all bombs and missiles are moved. On alternate times the player's gun base is moved.

PROCWDW: Draw aliens in starting positions at the start of each attack wave.

PROCZBP: Set all bomb positions to zero (off the screen).

PROCBLOCK: Draw bunkers for player's gun base to hide behind.

PROCMB: Test keyboard and move gun base & fire gun when keys are pressed.

PROCBOMBS: Move bombs and missiles and check for impact.

PROCMONSTER: Move one alien after making the following checks. If it is dead then do not bother with it. If it has been hit then finish it off. Otherwise move it.

PROCEXP: Explosion at position X,Y and sound effects of pitch P.

PROCDESTROY: When alien hit by missile this marks it for finishing off by PROCMONSTER and also updates the score.

PROCCALC: Calculates relative position of an alien in the block of aliens.


```

10MODE2
20COLOUR133
30CLS
40COLOUR135
50COLOUR0
60PRINTTAB(3,7)"  INVADERS  "
70COLOUR133
80PRINT/" "<" & ">" TO MOVE  BASE"
90PRINT/" "F" TO FIRE"
100PRINT/" "PRESS SPACE BAR TO  START"
110PRINTGET
120MODE2
130REM Define characters
140VDU23,224,66,60,126,219,126,102,153,66
150VDU23,225,0,8,20,8,8,8,0
160VDU23,226,0,8,8,8,8,20,8,0
170VDU23,227,4,4,126,255,255,255,255,126
180VDU23,228,32,32,126,255,255,255,255,126
190VDU23,229,0,0,24,60,60,24,0,0
200VDU23,230,0,60,126,126,126,126,60,0
210VDU23,231,126,195,129,129,129,129,195,126
220REM Envelopes for sound effects
230ENVELOPE1,1,-2,4,-2,4,4,4,34,-1,1,-1,100,70
240ENVELOPE2,130,-4,1,1,1,200,200,126,-7,-1,-1,
126,90
250ENVELOPE3,1,0,0,0,1,1,1,126,-2,0,-1,100,50
260DIMHZ(19),BPZ(5,1),PZ(19,1)
270REM Calculate relative positions of the horr
ible green things.
280PROCCALC
290REM Screen boundaries for the aliens.
300HIX=896
310X1Z=128
320X2Z=1058
330REM BOK = base OK
340LIFE=3
350BOK=TRUE
360REM Define graphics area
370VDU24,0;96;1279;1000;
380SCZ=0
390REM score

```

```

400REM Main loop
410REPEAT CLS
420CLZ=0
430REM Clock used to make base move only altern
ate times
440XZ=X1Z+16
450CZ=1
460DXZ=32
470DYZ=0
480XB=608
490YZ=H1Z-384
500EOLZ=FALSE
510CZ=0
520LSZ=FALSE
530MIXZ=0
540MIYZ=9999
550REM Set up frame of invaders
560PROCW
570PROCZBP
580PROCBLOCK
590INV=20
600REM Keep CZ in range 0-19.
610CZ=CZMOD20
620IFCLZANDBOK THENPROCMB
630CLZ=NOTCLZ
640PROCBOMBS
650PROCMONSTER
660REM Move T H E M !
670CZ=CZ+1
680IFCZ=20THENXZ=XZ+DXZ:YZ=YZ+DYZELSE610
690VDU4
700PRINTTAB(2,31);"SCORE";SCZ;"      "LIFE;
710SOUND2,1,5,5
720IFINV=0ORLIFE=0THENB00
730DYZ=0
740IFEOLZANDNOTLSZTHENDYZ=-64:DXZ=-DXZ
750LSZ=EOLZ
760EOLZ=FALSE
770IFNOTBOK THENLIFE=LIFE-1:BOK=LIFE>0:XB=608:G
OTO610
780GOTO610
790REM Clear key buffer.

```

```

800*FX15,1
810CLG
820HIZ=HIZ-64
830REM Advance next attack wave down screen
840UNTILLIFE=0
850REM You now dead.
860COLOUR2
870PRINTTAB(3,8)"Little green      monsters ru
le OK"
880END
890
900DEFPROC DW LOCAL IZ
910REM Draw them in initial positions
920GCOL0,2
930FOR IZ=0 TO 19
940M%(IZ)=TRUE
950MOVE(IZ*MOD5)*128+X1Z+16,HIZ-(IZ*DIV5)*128
960VDU5,224,4
970NEXT
980ENDPROC
990
1000DEFPROC ZBP LOCAL IZ
1010REM Zero bomb positions
1020FOR IZ=0 TO 5
1030BP%(IZ,0)=0
1040BP%(IZ,1)=0
1050NEXT
1060ENDPROC
1070
1080DEFPROC BLOCK
1090REM Draw bunckers
1100GCOL0,4
1110FOR IZ=0 TO 2
1120PZ=IZ*390+128
1130MOVE PZ,150
1140MOVE PZ,260
1150PLOT85,PZ+280,150
1160PLOT85,PZ+280,260
1170NEXT
1180ENDPROC
1190
1200DEFPROC MB LOCAL KZ
1210REM Move base & fire hyper laser cannon (gun

```

```

).
1220KZ=XB
1230IFINKEY(-103)ANDXB>64THENXB=XB-32
1240IFINKEY(-104)ANDXB<1146THENXB=XB+32
1250IFINKEY(-68)ANDMIYZ>1000THENMIXZ=XB+32:MIYZ=
128:SOUND0,2,45,2
1260GC0L0,0
1270MOVEKZ,128
1280VDU5,227,228
1290MOVEXB,128
1300GC0L0,3
1310VDU227,228,4
1320ENDPROC
1330
1340DEFPROCBOBMS LOCAL IZ,HZ
1350REM Move bombs & missiles etc
1360IFMIYZ>1000THEN1470
1370MOVEMIXZ,MIYZ
1380GC0L0,0
1390VDU5,226
1400MIYZ=MIYZ+32
1410GC0L0,7
1420HZ=POINT(MIXZ+24,MIYZ-8)
1430IFHZ=0THENMOVEMIXZ,MIYZ:VDU226:GOTO1470
1440PROCEXP(MIXZ,MIYZ,4)
1450IFHZ=2THENPROCDESTROY
1460MIYZ=9999
1470FORIX=0TO3
1480IFBPZ(IX,1)<32THENNEXT:ENDPROC
1490GC0L0,0
1500AZ=BPZ(IX,0)
1510BZ=BPZ(IX,1)
1520MOVEAZ,BZ
1530VDU5,225
1540HZ=POINT(AZ+24,BZ-32)
1550BPZ(IX,1)=BZ-24
1560AZ=AZ+4*SGN(XB+32-AZ)
1570BPZ(IX,0)=AZ
1580MOVEAZ,BZ-24
1590GC0L0,6
1600VDU225,5

```

```

1610IFHZ=0THENNEXT:ENDPROC
1620IFHZ=4THENPROCEXP(BPZ(IZ,0),BPZ(IZ,1),5):BPZ
(IZ,1)=0
1630IFHZ=3THENBOK=FALSE:PROCEXP(XB,128,6):PROCEX
P(XB+64,128,6):PROCEXP(XB+48,128,6)
1640NEXT
1650ENDPROC
1660
1670DEFPROCMONSTER LOCAL XMZ,YMZ
1680REM Move monsters
1690IF(CZAND3)=0THENFZ=TRUE
1700REM Check if monster has been zaped
1710IFMZ(CZ)=0ANDCZ=19THENENDPROC
1720IFMZ(CZ)=0THENCZ=CZ+1:GOTO1690
1730XMZ=XZ+PZ(CZ,0)
1740YMZ=YZ+PZ(CZ,1)
1750MOVEXMZ,YMZ
1760GCOLOR,0
1770VDU5,224
1780IFMZ(CZ)=1THENMZ(CZ)=0:PROCEXP(XMZ,YMZ,4):EN
DPROC
1790GCOLOR,2
1800MOVEXMZ+DXZ,YMZ+DYZ
1810VDU224,4
1820REM Check if it has reach then edge of the s
creen
1830EOLZ=EOLZORXMZ>=X2ZORXMZ<=X1Z
1840REM If first one in columb then try & find a
bomb for it to drop
1850IFFZTHENBZ=RND(4)-1:IFBPZ(BZ,1)<32THENBPZ(BZ
,0)=XMZ+16:BPZ(BZ,1)=YMZ-32
1860FZ=FALSE
1870IFYMZ<192THENLIFE=0
1880ENDPROC
1890
1900DEFPROCEXP(X,Y,P) LOCALIZ
1910REM Explosion & sound effect
1920GCOLOR,5
1930SOUND0,3,P,2
1940FORIZ=0TO11
1950MOVEX,Y
1960VDU229+IZMOD3

```

```

1970NEXT
1980GCOL0,0
1990FORIZ=0T03
2000MOVEX,Y
2010VDU229+IZMOD3
2020NEXT
2030ENDPROC
2040
2050DEFPROCDESTROY LOCALUX,VZ,DZ
2060REM Find which invader was hit & mark it
2070UZ=(MIXZ-XZ+48)DIV128
2080VZ=(MIYZ-YZ+48)DIV128
2090DZ=VZ+UZ*4
2100IFMZX(DZ)=TRUE THENINV=INV-1
2110MZ(DZ)=1
2120SCZ=SCZ+1
2130ENDPROC
2140
2150DEFPROCCALC
2160REM Calculate relative pos of invaders
2170FORCZ=0T019
2180PZ(CZ,0)=(CZAND-4)*32
2190PZ(CZ,1)=(CZAND3)*128
2200NEXT
2210ENDPROC

```

ANIMALS

DESCRIPTION

This game uses a data base of a tree-like structure to store questions and the names of animals. When playing the game, you have to think of an animal and the computer then has to find its name, by asking you questions. If it fails, you then tell it the name of the animal and a question that can be answered yes or no to distinguish it from the computer's best guess. The computer stores this information and the next time the game is played, it knows that animal.

The main program (lines 10-200) puts the computer in a teletext mode and displays the title using PROCHEADING. It also sets up arrays and initial data bases using PROCINIT. It then gives an option to load a data base using PROCCHOISE. The game is then played until the player says "NO" to "play again". An option to save the data base is then given before the program ends.

PROCEDURES AND FUNCTIONS USED

PROCHEADING:

Prints up title and instructions if asked.

FNYES:

Asks "(Y OR N)" until it gets an answer beginning with one of those two letters. If it is "Y" then the value is returned, otherwise false is returned.

PROCINIT:

Dimensions the array to contain the data base to size S% and reads initial data from data statements.

PROCCHOISE:

This gives a choice to load or to save data to the current filing system.

PROCPPLAY:

This starts off the game by asking the first question ("HAS IT GOT LEGS?"). This is done by taking the first question from the data base, combining it with "FNQ("and")" so that E\$ contains "FNQ("HAS IT GOT LEGS",1,2)". The 1 and 2 are the two pointers to the next data records for the question being answered yes or no (If the question is answered yes, then use data record 1.). FNQ is actually called by using PRINT EVAL on E\$.

FNQ:

This is the function that asks the questions. According to what the answer is, it branches into the data base, using the two pointer

in the parameters YES and NO. The variable NEX is the pointer to the next record in the data base (A\$(NEX)) to be used. The first letter of this record is either "Q" or "A" and is used as the name of the next function to be called with the parameters contained in the rest of the record (using EVAL again). So FNQ either calls itself or FNA.

FNA:

First ask the question "IS IT A" if the answer is yes then it returns with the value "I GOT IT!" and FNQ returns this to PROCPLAY to be printed. Otherwise if it has room in the data base, it asks for the name of the animal and a question about it. After adding it to the data base it returns with the value "I'll know that one next time."

PROCSAVE:

Saves the data base to the current filing system.

PROCLOAD:

Loads a data base from the current filing system.

Note the records in the data statements at the end of the program are enclosed in an extra pair of quotes. This is because each record has embedded commas.


```

10REM Animals game
20
30MODE7
40PROCHEADING
50PROCINIT(300)
60PROCCHOISE
70
80REM Main loop
90
100REPEAT
110PROCPLAY
120PRINT"" "Do you want to play again"
130UNTIL NOT FNYES
140
150REM Save or load data base
160
170PROCCHOISE
180PRINT"Do you want to play again"
190IFFNYES THEN100
200END
210
220
230
240DEFPROCHEADING
250PRINT""
260FORIX=0TO1
270VDU133+IX
275
280REM Double height
285
290VDU141
300PRINTTAB(10)"A N I M A L S"
310NEXT
320PRINT"" " Do you want instructions";
330IFNOTFNYES THENENDPROC
340PRINT"" " This is the game of animals.You choose an animal & I have to ask you some questions to find the name of the"
350PRINT"Animal you have chosen."" please answer my questions YES or NO."
360ENDPROC
365

```

```

370
380DEFFNYES LOCAL A$
385
390REM Returns true false for yes or no
395
400INPUT" (Y OR N) "A$
410IFLEFT$(A$,1)="Y"THEN=TRUE
420IFLEFT$(A$,1)="N"THEN=FALSE
430GOTO400
440
445
450DEFPROCINIT(SZ)
460DIMA$(SZ)
465
470REM Get initial data
475
480`=SZ
490FORI=0TO2
500READA$(I)
510NEXT
520FR=I
530ENDPROC
540
550DEFPROCCHOISE
555
560PRINT"Do you wish to load or save data to ta
pe"
565
570IF NOT FNYES THENENDPROC
580PRINT"LOAD A DATA FILE ? ";
590IFFNYES THENPROCLOAD:ENDPROC
600PRINT"SAVE DATA ? ";
610IFFNYES THENPROCSAVE
620ENDPROC
625
630
640DEFPROCPLAY
650PRINT"" Think of an animal ""
660E$="FNQ("+A$(0)+")"
670REM Start with FNQ("HAS IT GOT LEGS",1,2)
680PRINTEVALE$
690ENDPROC

```

```

695
700
710DEFFNQ(Q$,YES,NO)
715
720REM Ask question about your animal
725
730PRINTQ$
740IFFNYES THEN NEX=YES ELSE NEX=NO
750S$=LEFT$(A$(NEX),1)
760R$=MID$(A$(NEX),2)
770E$="FN"+S$+"("+R$+"")"
780=EVALE$
785
790
800DEFFNA(A$)
805
810REM Has come to a name of an animal in the d
ata
815
820PRINT"Is it a ";A$;" ? "
830IFFNYES THEN="I GOT IT !"
840IFFR+2>` THEN="SORRY I HAVE NO MORE ROOM FOR
ANIMALS"
845
850REM Add new animal to data
855
860TEMP$=A$(NEX)
870PRINT"What was your animal"
880INPUTAN$
890PRINT"A question about A difference between
the two."
900INPUTQ$
910PRINT"What would the answer be for a"AN$
914
915REM Store new question & then the two animals
it is about.
916
920AN$="A"+CHR$34+AN$+CHR$34
930IFFNYES THEN T1$=TEMP$:TEMP$=AN$:AN$=T1$
940A$(FR)=AN$
950A$(FR+1)=TEMP$

```

```

960A$(NEX)="Q"+CHR$34+Q$+CHR$34+"", "+STR$(FR+1)+
", "+STR$FR
970FR=FR+2
980="I'll know that one next time."
990
1000
1010DEFPROC SAVE LOCAL C, IZ
1020C=OPENOUT("ANIMAL_D")
1030PRINT#C, FR-1
1040FOR IZ=0 TO FR-1
1050PRINT#C, A$(IZ)
1060NEXT
1070CLOSE#C
1080ENDPROC
1085
1090
1100DEFPROC LOAD LOCAL C, IZ
1110C=OPENIN("ANIMAL_D")
1120INPUT#C, FR
1130FOR IZ=0 TO FR
1140INPUT#C, A$(IZ)
1150NEXT
1160CLOSE#C
1170FR=FR+1
1180ENDPROC
1190
1195
1200DATA "" "HAS IT GOT LEGS"", 1, 2, "A" "LION" ""
1210DATA "A" "FISH" ""
>

```

BATTLE SHIPS

DESCRIPTION

This is usually a two player game played with pencil and paper. Each player marks out a ten by ten grid on his paper. They then mark the position of their ships on the grid as lines of horizontal or vertical squares, each type of ship being so many squares long. The players then take turns at guessing the position of the other player's ships in order to sink them. In response to the opponent's guess, say "2,3", the other player must say "hit" or "missed" or "you sunk my" when the ship has been hit. When replying to a player's guess be very careful to reply correctly if you do not want to be accused of cheating.

All of the above applies to this version, except that one player is the computer. The computer performs two tasks; one, it plays as the opponent and two, it acts as the player's pencil and paper. The two tasks are kept separate and the part of the program that calculates the computer's move never cheats by looking at your ships. However the computer may accuse you of cheating and take appropriate action.

The main program is from line 10 to line 200. It puts the computer in teletext mode and dimensions the arrays used. The arrays used are:

S:

This is used as four ten by ten grids to hold the positions of the computer's ships, the player's ships, the markers for where the computer has tried shooting at and the same applies to the player.

NAMES\$:

This holds the names of the different types of ships in order of length in squares.

DI:

Holds four directional pointers used when the computer is calculating its move.

The main program that prints the introduction gets the player's and the computer's ships in place, plays the game, comments on the result and repeats if required.

THE FUNCTIONS AND PROCEDURES USED ARE:

PROC COMPUTER INIT:

This initialises the values of all the arrays and then randomly positions the computer's ships.

PROC PLAYER INIT:

This positions the player's ships using PROC SHIP.

PROCCSHIP:

Randomly chooses whether a computer's ship of length L is to

be positioned horizontally or vertically.

FNCVSHIP:

This randomly places a computer's ship of length L vertically. Returns with the value true, if space for the ship was found.

FNCHSHIP:

Randomly places a computer's ship of length L horizontally. Returns with the value true, if space for the ship was found.

PROCPSHIP:

Ask for the position of a player's ship of length L and whether it is to be horizontal or vertical.

PROCPSHOW:

Show positions of player's ship and whether they have been hit.

FHHPSHIP:

Try to position player's ship of length L horizontally at a given position and return with the value true if it can be done.

FNVPSHIP:

Try to position player's ship of length L vertically at a given position and return with the value true if it can be done.

PROCPLAY:

Give player and computer alternate goes at firing at each other, until one of them has no ships left.

PROCPLAYER:

Player's go. Show where player has tried before and ask for next guess. Then report on the result of his shot.

PROCSHOWT:

Print up a grid showing the results of the player's guesses so far "X" for miss and "*" for hit.

PROCSUNK:

Check to see if the player has sunk a ship belonging to the computer.

FNS:

This function is to read the grid showing the computer's past guesses.

PROCCOMPUTER:

This calculates or guesses the computer's go. HP is the number of hits it has got that have not been accounted for by a ship being sunk. If HP is zero then just randomly choose a point that has not been tried before. Otherwise look at the pattern of unaccounted for

hits and use as a guide to the next shot.

PROCRAND:

Just find a place that has not been tried before by the computer.

PROCTRY:

Ask the player if the computer's guess calculated by PROCRAND or PROCINTEL is a hit, a miss or a sinking. Then update grid containing computer's guesses accordingly.

PROCSD:

Move the computer's search pointers to its guesses grid in direction D.

PROCINTEL:

This is to find an untried position at the end of a line of hits. It first looks for a line of hits in any direction from the position of the first hit. If this fails then it searches from the position of the last hit. If it can still not find a line of hits with an untried position at the end, then try random searches about first and last hits. If this fails then assume the player replied falsely to the question "hit, miss or sinking" and take appropriate action (PROCATOMIC).

PROCHITS:

Search in all directions with PROCHITSI.

PROCHITSI:

Search in direction D for an untested position at the end of a line of hits.

PROCRAb:

Search for an untried position around point U.V.

PROCGOING DOWN:

When the computer sinks a player's ship, the computer is no longer interested in those hit markers, so they are changed to position tried markers.

PROCCI:

Search in direction 1 for a line of hits of length L. If found, then change them to position tried.

PROC COMMENT:

Say who won etc.

FYNES:

Ask question "(Y/N)" return true if the answer is yes.

PROCINTRO:

Print introduction and instructions.

PROCATMOIC:

Get nasty if player cheats.


```

10 REM*****
20 REM***** Battle Ships *****
30 REM*****
40
50 MODE7
60
70 REM S(x,x,0) is position of computers ships
80 REM S(x,x,1) is position of players ships
90 REM S(x,x,2) is where computer has tried
100 REM S(x,x,3) is where player has tried
110
120 DIMS(9,9,3),NAMES$(7),DI(3,1)
130 REPEAT
140     PROC_INTRO
150     PROC_COMPUTER_INIT
160     PROC_PLAYER_INIT
170     PROC_PLAY
180     PROC_COMMENT
190     UNTILNOTFNYES("ANOTHER GAME")
200 END
210
220
230 DEF PROC_COMPUTER_INIT
240
250 REM Set array S( ) to zeros
260
270 FORI=0TO9
280     FORJ=0TO9
290         FORK=0TO3
300             S(I,J,K)=0
310         NEXT
320     NEXT
330 NEXT
340
350 REM Read search dirrection vectors
360
370 RESTORE
380 FORI=0TO3
390     READDI(I,0),DI(I,1)
400 NEXT
410 REM Read in names of ship types
420

```

```

430 FORI=1TO7
440   READNAMES$(I)
450   NEXT
460
470 REM Number of hits that have not sunk any s
hips
480
490 HP=0
500
510 REM Position computer's ships
520
530 FORI=7TO1STEP-1
540   PROCCSHIP(I)
550   NEXT
560 PRINT''' " All my ships are in place."
570 ENDPROC
580
590
600 DEF PROC_PLAYER_INIT
610
620 REM Position player's ships
630
640 PRINT''' " You may now position your ships &
I   promise I will not look."
650 FORI=7TO1STEP-1
660   PROCPSHIP(I)
670   NEXT
680 ENDPROC
690
700
710 DEF PROCCSHIP(L) LOCAL OK
720
730 REM Position a computer's ship of length L
740
750 REPEAT
760   IFRND(2)=1THENOK=FNCVSHIP(L) ELSEOK=FNCHS
HIP(L)
770   UNTIL OK
780 ENDPROC
790
800

```

```

810 DEF FNCVSHIP(L) LOCAL X,Y,C,I
820
830 REM Try to position a computer's ship of le
ngth L vertically
840
850 X=RND(10)-1
860 Y=RND(10-L)-1
870 FORI=0TOL-1
880   C=C+S(X,Y+I,0)
890   NEXT
900 IF C<>0 THEN=FALSE
910 FORI=0TOL-1
920   S(X,Y+I,0)=L
930   NEXT
940 =TRUE
950
960
970 DEF FNCHSHIP(L) LOCAL X,Y,C,I
980
990 REM .... horizontaly
1000
1010 X=RND(10-L)-1
1020 Y=RND(10)-1
1030 FORI=0TOL-1
1040   C=C+S(X+I,Y,0)
1050   NEXT
1060 IF C<>0 THEN=FALSE
1070 FORI=0TOL-1
1080   S(X+I,Y,0)=L
1090   NEXT
1100 =TRUE
1110
1120
1130 DEF PROCPSHIP(L) LOCAL X,Y,OK
1140
1150 REM Position player's ship of lenght L
1160
1170 REPEAT
1180   PROCPSHOW
1190   PRINT" POSITION OF "NAMES$(L);
1200   INPUT" (X,Y) : "X,Y

```

```

1210 REPEAT
1220 IFL<>1 THENINPUT"POSITION IT HORIZONTAL
LY OR VERTICALLY",A$ ELSEA$="H"
1230 A$=LEFT$(A$,1)
1240 UNTILA$="H"ORA$="V"
1250 IFA$="H"THENOK=FNHPSHIP(L) ELSE OK=FNVPSH
IP(L)
1260 UNTIL OK
1270 PRINT"" OK"
1280 ENDPROC
1290
1300
1310 DEF PROCPSHOW LOCALI,J
1320
1330 REM Show player ship positions
1340
1350 ^=INKEY(400)
1360 CLS
1370 PRINTSTRING$(40,"_");
1380 PRINTCHR$129" 0 1 2 3 4 5 6 7 8
9"
1390 FORI=0TO9
1400 PRINTCHR$129;I;
1410 FORJ=0TO9
1420 V=S(J,I,1)
1430 C=130
1440 IFV=0THENC=132
1450 IFV>=8THENC=133
1460 VDU137,C
1470 PRINT;7ANDV;
1480 NEXT
1490 PRINT
1500 NEXT
1510 PRINTSTRING$(40,"_");
1520 ENDPROC
1530
1540
1550 DEF FNHPSHIP(L) LOCAL I,C
1560
1570 REM Try to place it horizontally
1580
1590 IFX<0ORX+L>10ORY<0ORY>9THEN=FALSE

```

```

1600 FORI=0TOL-1
1610   C=C+S(X+I,Y,1)
1620   NEXT
1630   IFC<>0THEN=FALSE
1640   FORI=0TOL-1
1650     S(X+I,Y,1)=L
1660     NEXT
1670   =TRUE
1680
1690
1700 DEF FNVPSHIP(L) LOCAL I,C
1710
1720 REM Try to position it vertically
1730
1740 IFX<0ORX>9ORY<0ORY+L>10THEN=FALSE
1750 FORI=0TOL-1
1760   C=C+S(X,Y+I,1)
1770   NEXT
1780   IFC<>0THEN=FALSE
1790   FORI=0TOL-1
1800     S(X,Y+I,1)=L
1810     NEXT
1820   =TRUE
1830
1840
1850 DEF PROC_PLAY
1860
1870 REM Let computer & player have alternate go
es at blasting each other
1880
1890 PRINT"/" Now let's start the battle."
1900 AL=TRUE
1910 TC=28
1920 TP=28
1930 REPEAT
1940   IFAL THENPROCPLAYER ELSE PROCCOMPUTER
1950   AL=NOTAL
1960   UNTILTC=0 ORTP=0
1970 ENDPROC
1980
1990

```

```

2000 DEF PROCPLAYER LOCALX,Y,P
2010
2020 REM Players go
2030
2040 PROCPSHOWT
2050 REPEAT
2060   INPUT"Your go."/"Position X,Y : "X,Y
2070   UNTILX>=0 ANDY>=0 ANDX<10 ANDY<10
2080   P=S(X,Y,0)<>0AND0=(S(X,Y,0)AND8)
2090   S(X,Y,3)=1-P
2100   S(X,Y,0)=S(X,Y,0)-P*8
2110   IFP THENPRINT/"BANG !"/" A HIT! " ELSEPRIN
T/"SPLASH"
2120   TC=TC+P
2130   IFP THENPROCSUNK
2140   ENDPROC
2150
2160
2170 DEF PROCPSHOWT LOCAL I,J,V
2180
2190 REM Show player's target field
2200
2210   ^=INKEY(300)
2220   CLS
2230   PRINTSTRING$(20,"+ _");
2240   PRINT"    0  1  2  3  4  5  6  7  8  9"
2250   FORI=0TO9
2260     PRINT;I;
2270     FORJ=0TO9
2280       V=S(J,I,3)
2290       IFV=0THENPRINT"  .";
2300       IFV=1THENPRINT"  X";
2310       IFV=2THENPRINT"  *";
2320     NEXT
2330   PRINT
2340   NEXT
2350   PRINTSTRING$(20,"+ _");
2360   ENDPROC
2370
2380
2390 DEF PROCSUNK LOCALV,C,I,J

```

```

2400
2410 REM Check if player has sunk one of the com
puter's ships"
2420
2430 V=S(X,Y,0)
2440 FORI=0TO9
2450   FORJ=0TO9
2460     IFS(I,J,0)=V THENC=C+1
2470     NEXT
2480   NEXT
2490 IFC>(V AND7)THENPRINT"ERROR":STOP
2500 IFC<(V AND7)THENENDPROC
2510 PRINTCHR$131"You sunk my "NAMES$(V AND7)".
2520 ENDPROC
2530
2540
2550 DEF FNS(X,Y)
2560
2570 REM Read computer's target field. Return 1
(tried there before) if position is outside of ar
ray
2580
2590 IFX<0ORY<0ORX>9ORY>9THEN=1
2600 =S(X,Y,2)
2610
2620
2630 DEF PROCCOMPUTER LOCALX,Y
2640
2650 REM Computer's go
2660
2670 IFHP=0 THEN PROCRAND ELSE PROCINTEL
2680 PROCTRY
2690 ENDPROC
2700
2710
2720 DEF PROCRAND LOCAL AT
2730
2740 REM Random shot about X,Y
2750
2760 SD=RND(4)-1
2770 REPEAT
2780   X=RND(10)-1

```

```

2790   Y=RND(10)-1
2800   AT=AT+1
2810   UNTILS(X,Y,2)=0 OR AT>1500
2820 IFAT>150 THENPROC__ATOMIC
2830 ENDPROC
2840
2850
2860 DEF PROCTRY
2870
2880 REM Try a shot at the calculated position
2890
2900 PROCPSHOW
2910 PRINTTAB(X*3+2,Y+2)CHR$136
2920 PRINTTAB(0,13);
2930 PRINTX,Y
2940 REPEAT
2950   INPUT"Was that a hit a miss or a sinking"
  ✓"(H,M,S) ",A$
2960   UNTILA$="H"ORA$="M"ORA$="S"
2970   S(X,Y,1)=S(X,Y,1)+8
2980   IFA$="M"THENS(X,Y,2)=1:ENDPROC
2990   TP=TP-1
3000   S(X,Y,2)=2
3010   LHX=X
3020   LHY=Y
3030   IFHP=0THENFHX=X:FHY=Y
3040   HP=HP+1
3050   IFA$="S"THENPROCGOING_DOWN:ENDPROC
3060 ENDPROC
3070
3080
3090 DEF PROCSD(D)
3100
3110 REM Move search position in dirrection SD
3120
3130 D=(D+8)MOD4
3140 X=X+DI(D,0)
3150 Y=Y+DI(D,1)
3160 ENDPROC
3170
3180

```



```

3190 DEF PROCINTEL LOCAL SD,T
3200
3210 REM Find a untried position at the end of a
line of hits. Start searching for pos of first h
it. If that fails then search from last hit posit
ion
3220 REM If that fails then try a random shot ab
out first if that fail then last hit pos. If no l
uck by now the player must have replied wrong to
(h,n,s) get nasty.
3230
3240 SD=RND(4)-1
3250 PROCHITS(FHX,FHY)
3260 IFT THENENDPROC
3270 PROCHITS(LHX,LHY)
3280 IFT THENENDPROC
3290 PROCRAH(FHX,FHY)
3300 IFT THENENDPROC
3310 PROCRAH(LHX,LHY)
3320 IFT THENENDPROC
3330 PROC__ATOMIC
3340 ENDPROC
3350
3360
3370 DEF PROCHITS(U,V) LOCAL I
3380
3390 REM Search around U,V for a line of for a u
ntested position at the end of a line of hits.
3400
3410 FORI=0TO3
3420   X=U
3430   Y=V
3440   PROCSB(I+SD)
3450   IFFNS(X,Y)=2THENPROCHITS1(I+SD)
3460   NEXT
3470 ENDPROC
3480
3490
3500 DEF PROCHITS1(D)
3510
3520 REM Search in dirrection D for a line of fo

```

r a untested position at the end of a line of hits.

```
3530
3540 REPEAT
3550   PROCSD(D)
3560   UNTILFNS(X,Y)<>2
3570   IFFNS(X,Y)=0THENT=TRUE:SD=D:I=3
3580 ENDPROC
3590
3600
3610 DEF PROCRAE(U,V) LOCALI
3620
3630 REM Random search about U,V for a untried position
3640
3650 FORI=0TO3
3660   X=U
3670   Y=V
3680   PROCSD(I+SD)
3690   IFFNS(X,Y)=0THENT=TRUE:I=3
3700   NEXT
3710 ENDPROC
3720
3730
3740 DEF PROCGOING_DOWN LOCALI,C,A,T
3750
3760 REM When computer has sunk a ship change hit markers to tried (2 to 1)
3770
3780 REPEAT
3790   INPUT"What did I sink (1-7) ",A
3800   UNTILA>0 ANDA<8
3810   PRINT"I GOT YOUR "NAMES$(A)!"
3820   FORI=0TO3
3830     X=LHX
3840     Y=LHY
3850     PROCC1(X,Y,A-1)
3860     IFT THENI=3
3870     NEXT
3880   IFNOTT THENSTOP
3890   S(X,Y,2)=1
3900   HP=HP-A
```

```

3910 IFHP=0THENENDPROC
3920 T=FALSE
3930 FORI=0TO9
3940   FORJ=0TO9
3950     IFS(I,J,2)=2THENLHX=I:LHY=J:FHX=I:FHY=J
:T=TRUE
3960   NEXT
3970 NEXT
3980 IFNOT T THENSTOP
3990 ENDPROC
4000
4010
4020 DEF PROCC1(X,Y,L)
4030
4040 REM Recursive search in direction I for a l
ine of hits of length L .If found change to pos t
ried on exit. (See text.)
4050
4060 PROCSD(I)
4070 T=(L=0)
4080 IFFNS(X,Y)<>2ORT THENENDPROC
4090 PROCC1(X,Y,L-1)
4100 IFT THENS(X,Y,2)=1:ENDPROC
4110 ENDPROC
4120
4130
4140 DEF PROC_COMMENT
4150
4160 REM Message at end of game
4170
4180 IFTP=0THENPRINT'"I win..!!" ELSEPRINT'"I
will have my revenge."
4190 IFNOT FNYES(" Do you wish to see the positi
ons of my ships")THENENDPROC
4200 FORI=0TO9
4210   FORJ=0TO9
4220     S(I,J,1)=S(I,J,0)
4230   NEXT
4240 NEXT
4250 PROCPSHOW
4260 ENDPROC

```

```

4270
4280
4290 DEF FNYES(A$)
4300
4310 REM Ask yes or no
4320
4330 PRINTA$;
4340 REPEAT
4350   INPUT" (Y/N)",A$
4360   A$=LEFT$(A$,1)
4370   UNTILA$="Y"ORA$="N"
4380   =A$="Y"
4390
4400
4410 DEF PROC_INTRO
4420
4430 REM Introduction & instructions
4440
4450 CLS
4460 PRINT'''
4470 FORI=0TO1
4480   VDU132,157,141,134+I
4490   PRINT"   B A T T L E   S H I P S   "CHR
$(156)
4500   NEXT
4510 PRINT'" The computer will first position i
ts   own ships. Then ask you for the   posi
tions of your ships."
4520 PRINT'" When all the ships are in position
thenthe game begins. The object of the game is to
sink all the computer'sships   before it sink
s yours. The only problem being you do not know w
here each other'sships are."
4530 PRINT'" Positions for shots are for example
3,6 which is 3 across & 6 down."'" PRESS AN
Y KEY TO CONTINUE : "GET$
4540 CLS
4550 PRINT'"CHR$134"   BATTLE SHIPS"CHR$134"
"
4560 PRINT'"'" When calculating it's move the co
mputerdoes not look at the positions of your shi

```

```

ps at all, so you will have to tell it if it has
a hit ,a miss or sunk your ship. ";
4570 PRINT"Be very careful not to lie, as thecom
puter will find you out."////" PRESS ANY KEY TO
PLAY : "GET$
4580 CLS
4590 ENDPROC
4600
4610
4620 DEF PROC__ATOMIC
4630
4640 REM get nasty
4650
4660 CLS
4670 PRINT////" I can't find the rest of this shi
p."////" I think you LIE !"////CHR$129CHR$136"
  BYE"
4680 PRINT////" Note for observers :"/" The comp
uter just dropped a H bomb"/"on this fellow's shi
ps."
4690 END
4700
4710
4720
4730 REM X,Y of the four directions (0-3)
4740
4750 DATA 0,1,1,0,0,-1,-1,0
4760
4770 DATADINGHY,PATROL BOAT,SUB,FRIGATE,CRUISER,
BATTLE SHIP,CARRIER
>

```

MUTANT

This adventure game was not originally written for the BBC Micro but has now been fully converted to produce multi-coloured text in mode 7. The game is fully text based and gives detailed descriptions of the caverns, caves, tunnels and halls that lie under the surface of the planet Mutant. In case you are wondering why you should be under the surface of the planet I shall explain.

You are in your last year of space training college taking your second level test. This is a computer simulation of what it would be like to be alone after your space ship crashed on one of the most dangerous planets known. The planet Mutant is dangerous because of its radiation level, after a few hours you start to mutate into something like the other things that walk the caves of the planet. The test starts after you have made your first mistake, you climbed into a pit and you are now below the surface of the planet. All you have to do now is survive until you find a space ship. The only help you were given was that you would need the dark rod of muton to escape the planet. Another problem you must contend with is that the planet Mutant was once the size of the earth but has now shrunk under the effects of its own radiation to the size of a small asteroid. This means you can walk around the planet in a very short time and this can be very confusing. Also the planet is constantly changing so each time you start this computer test the planet will have changed. Note: as this is a second level test you are not expected to escape first time.

The test computer understands 16 verbs; these are: GO, GET, TAKE, DROP, THROW, SHOOT, DROP, WAVE, LIGHT, HELP, OPEN, PUSH, QUIT, RUB, JUMP, INVENTORY and LOOK. When you refer to objects you do not need to give the computer the full description. For example GET TREE would have the same effect GET LARGE CARNIVOROUS TREE. Also GO NORTH can be abbreviated to N and the same applies to all other directions. Finally if you fail this test five times you may have to forget about Star Fleet and apply for a job with the Merchant Fleet.

DESCRIPTION

On being run, this program first makes sure the computer is in mode 7. It then dimensions its arrays. These are:

JO: This contains information as to which rooms are joined to which, and the codes for the exit descriptions between the rooms.

OB%: Holds the room number of each object or monster etc.

CO\$: This contains a list of command words (verbs) like "DROP", "GET", "N", "S" etc.

TBUFF%: Is the text buffer used in the Teletext colour printing routine "PROCP" and holds the current line of text being printed.

OB\$: This holds the list of object descriptions like "LAMP", "LASER", "STATUE OF A BAT" etc.

DO\$: This contains the exit descriptions. Descriptions of an

even number are the open version of the next one in DO\$. For example "a large hole in a steel panel" is the open version of "a steel panel".

WA\$: This contains the key words for the descriptions in DO\$. In the above example, the key words would be "STEEL PANEL". The key words are compared with the second part of the command line, to see if the command refers to an exit eg. "PUSH WEB".

DIS\$: This holds "NORTH", "SOUTH" etc.

During the pauses in printing the title and introduction, the program initialises the above arrays (initialisation routine 4050-4740). All the text for the arrays comes from the data statements and the required numbers are generated at random, subject to some conditions.

It then enters the main program via the inventory routine (line 3300), so that the objects being held are displayed. This then does the normal exit for verb routines to the look routine (line 520), thus displaying the current position and objects present.

Next the program enters the random events section (670-720). If you survive this, then the subroutine for delayed events (410-480) follows next. For instance, the count down to explosions or being eaten by something unfriendly. Then comes the input routine (760-110). The program accepts a line of text and strips off leading spaces. The line is then split into the first word and any remaining words. The first word is compared with the verb list and the rest of the line compared with the object descriptions. The number of items they matched with, is stored in the variables OB% and CO (single variables with the same name as the arrays they point into). No match found is indicated by the value -1. CO is used at line 1100 to branch to the right verb routine, when they are finished they re-enter by 520.

```

10REM Adventure game MUTANT
20REM For 32K BBC micro
30
40
50MODE7
60PRINT"////"
70CLEAR
80
90REM Double height
100
110PRINTTAB(9)CHR$(141)"Planet Of Mutant"
120PRINTTAB(8)CHR$(131);CHR$(141);"Planet Of Mu
tant"
130PRINTTAB(10)CHR$131"By C.N.Graham."
140DIMJO 612,OBX(30),CO$(21),TBUFZX 255
150FORIZ=0TO612
160JO?IZ=0
170NEXT
180REM JO is place joining data. lsb is type of
path msb is num of next room. OBX( ) is the room
containing the object. CO$( ) contains the comma
nd words.
190DIMOB$(30),DO$(15),WA$(7),DI$(5)
200 REM OB$( ) object name. RO$( ) Room descrip
tion. DO$( ) Door way discription. WA$( ) Key wor
ds for doors. DI$( ) NORTH EAST etc.
210PROCP(" THIS IS A SECOND LEVEL COMPUTER TEST
FOR STAR CRUISER CREW.IF YOU TAKE MORE THAN FIVE
ATTEMPTS TRY FOR A JOB IN THE MERCHANT FLEET.",1
34)
220
230REM Initialise
240
250GOSUB4050
260PRINT
270PROCP("Your ship a 'V' class star cruiser cr
ash landed on the unexplored planet MUTANT. This
planet has high levels of radiation of a type onl
y found here.",130)
280PROCP("This causes strange mutations in any
living thing trapped on the planet for more",130)
290PROCP("than a few hours. One hour has alread

```


y passed since some of the crew went to test the radiation level at the bottom of a near-by pit.", 130)

300PROCP("You are now the only crew member alive on the crippled ship so you have gone to find the men in the pit. But there is no sign of them & you can not get out of the pit without help",130)

310PRINT'CHR\$129"----- HELP "CHR\$136"!"

320

330

340

350GOSUB370

360GOTO3300

370A=INKEY(500)

380RETURN

390REM -----

400

410IFEX=0THEN440

420EX=EX-1

430IFEX=-3THEN1230

440IF00=0THENRETURN

450TH=TH+1

460IFRND(TH)>3THENOBZ=00:GOTO1190

470IFTH>4THENPROCP("I think the "+OB\$(00)+" wants you for a bite between meals.",129):PRINT

480RETURN

490FORI=0TO3000

500NEXT

510

520REM * * * LOOK routine

530

540IFPOS<>0THENPRINT

550IF(NOTLI OR(OBZ(0)<>-1ANDOBZ(0)<>RN))ANDRN<>0THENVDU132:PRINT"It's too dark I can not see anything":GOTO750

560PROCP("You are "+FNRO(RN)+".",133)

570FORI=0TO5

580DO=?FNJO(RN,I)

590IFDO=0THEN610

600IFI<2THENPROCP("There is "+DO\$(DO-2)+".",133)

```

) ELSEPROCP("To the "+DI$(I)+" there is "+DO$(DO-
2)+"",133)
610NEXTI
620IFPOS<>0THENPRINT
630FORI=0TO30
640IFOBZ(I)=RN THENPROCP(OB$(I)+" ",134)
650NEXT
660IFPOS<>0THENPRINT
670IFOBZ(21)=-1THENIFRND(5)=5THENVDU129:PRINT"A
big black snake came out of shadows &"CHR$129"sw
allowed you & the lump of meat in one"CHR$129"go.
"CHR$130"Try again.":GOTO60
680IFRN=49THENIFRND(7)=7THENPRINT"Swoosh Splat!
I think that sign said 'Danger falling rocks.'":
GOTO60
690IFRND(65)=15THENIFOBZ(23)=-1THENOBS(23)="RED
STICK":OBZ(23)=RN:PRINT"I think you dropped some
thing.":FORI=0TO500:NEXT
700IFRND(100)=100THENPRINT"zzzzZAP "CHR$136"! "C
HR$137" Oh dear I think some one thought you w
ere a mutant. "CHR$130" Try again.":GOTO60
710IFRN=50THENPRINTSTRING$(40,"*");" "CHR$
136"Well done."CHR$137"You escaped at last. Try a
gain see if you can do it quicker this time.":FOR
I=0TO5000:NEXT:GOTO60
720
730REM Input command line
740
750GOSUB410
760INPUTLINE": "C$
770IFC$=""THEN750
780
790REM Remove leading spaces
800IFLEFT$(C$,1)=" "ANDC$<>""THENC$=MID$(C$,2):
GOTO800
810IFC$=""THEN750
820L=INSTR(C$," ")
830IFL=0THENC1$=C$:C2$="":GOTO860
840C1$=LEFT$(C$,L-1)
850IFL<LEN(C$)THENC2$=RIGHT$(C$,LEN(C$)-L) ELSE
C2$=""

```

```

860IFLEFT$(C2$,1)<>" THEN880
870IFC2$<>" THENC2$=RIGHT$(C2$,LEN(C2$)-1):GOT
0860 ELSEC2$=""
88000=0
890FORI=11TO18
900IFI<>14ANDOBZ(I)=RN THEN00=I
910NEXTI
920I=0
930IFC1$=C0$(I)THENC0=I:GOTO960
940I=I+1
950IFI>21THENC0=-1ELSE930
960IFC0=-1THENPRINTCHR$129"I do not know how to
";C1$;:IFC2$<>"THENPRINT" something";
970IFC0=-1THENPRINT".":GOTO750
980I=30
990IFC2$=""THENOBZ=-1:GOTO1100
1000LP=FNINSTR(OB$(I),C2$)
1010IFLP THENOBZ=I:GOTO1040
1020I=I-1
1030IFI<0THENOBZ=-1ELSE1000
1040IFOBZ=-1THEN1100
1050IFOBZ(OBZ)<>RN ANDOBZ(OBZ)<>-1THEN02=-1ELSE1
100
1060FORJ=0TOI
1070IFFNINSTR(OB$(J),C2$)AND(OBZ(J)=-1OROBZ(J)=R
N)THEN02=J
1080NEXTJ
1090IF02<>-1THENOBZ=02
1100ONC0+1GOTO1130,1130,1590,1720,1910,2320,2400
,2500,2610,2740,3080,3250,3300,3380,3590,3640,369
0,3740,3790,3840,3890,490
1110
1120
1130REM * * * TAKE & GET routine
1140
1150IFC2$=""THENPRINT"What ?":GOTO750
1160IFOBZ=-1THENPRINT"You cannot do that":GOTO75
0
1170IFOBZ(OBZ)<>RN THENPRINT"I see no ";C2$;" he
re.":GOTO750
1180IFOBZ<10THEN1510
1190 ON OBZ-9 GOTO1270,1250,1250,1250,1230,1320,

```

```

1250,1320,1340,1510,1510,1510,1510,1220,1430,1450
,1470,1230,1510,1480,1510
1200PRINT" ERROR *****"
1210STOP
1220IFOB$(1)=-1THENPRINT"It weighs a ton I can't
lift it.":GOTO750 ELSEOB$(23)="BLACK STICK":GOT
O1510
1230PRINT" * * * B A N G * * * Oh well st
art again."
1240GOTO60
1250PROCP("Well at least the "+OB$(OB%)+ " is hap
py it has just eaten half of you.          Try agai
n.",129)
1260GOTO60
1270 PRINT"Oh ! I think I have just done my self
an injury. "
1280FORI=0TO900
1290NEXT
1300PRINT"   I have ! Let's start again."
1310GOTO60
1320PRINT"The ";OB$(OB%);" bit you .You are now
a mutant astronaut. I think maybe four arms & a
green scaly face suits you. Try again."
1330GOTO60
1340PRINT"You pricked your self ."
1350FORI=0TO200
1360NEXT
1370PRINT"You feel numb all over."
1380FORI=0TO200
1390NEXT
1400PRINT"Suddenly you go into a violent fit tha
t throws you backwards."
1410DI=BD
1420GOTO2840
1430PRINT"A ten foot crab comes from beneath the
water and grabs you so good bye.          Try again
."
1440GOTO60
1450PRINT"The edge of the pool collapses & you f
all in not to be seen again. You're not doing ve
ry well are you.          Try again."

```

```

1460GOTO60
1470IFOBZ(22)=-1THENPRINT"You know you can't dri
nk that you are in a space suit.":GOTO1510 ELSE1
490
1480IFOBZ(30)=-1THEN1510 ELSE1490
1490PRINT"Sorry you can't do that."
1500GOTO750
1510IT=0
1520FORI=0TO30
1530IFOBZ(I)=-1THENIT=IT+1
1540NEXT
1550IFIT>6THENPRINT"Sorry you can not carry any
more.":GOTO750
1560OBZ(OBZ)=-1
1570GOTO520
1580
1590REM * * * DROP routine
1600
1610IFC2$=""THENPRINT"Drop what ?":GOTO750
1620IFOBZ=-1THENPRINT"I do not remember picking
up a ";C2$;" in here.":GOTO750
1630IFOBZ(OBZ)<>-1THENPRINT"I'm not holding a ";
C2$:GOTO750
1640OBZ(OBZ)=RN
1650IFOBZ=28ANDOBZ(18)=RN THENPRINT"The green fl
y jumped on the rose bush. Then it started doing
what all good green flies do & that's the end of
the rose bush.":OBZ(18)=100:GOTO490
1660IFOBZ=19ANDOBZ(13)=RN THENPRINT"The wood pec
ker turned the tree into sawdust.":OBZ(13)=100:GO
TO490
1670 IFOBZ=20THENPRINT"B A N G .           Try
again.":GOTO60
1680 IFOBZ=29THENPRINT"It dug a hole by your fee
t so quick that you fell through it.":RN=RND(47)-
1:GOTO490
1690 IFOBZ=30ANDOBZ(29)=-1THENOBZ(29)=RN:GOTO520

1700GOTO520
1710
1720REM * * * THROW routine
1730

```

```

1740IFC2$=""THENPRINT"Throw what ?":GOTO750
1750IFOBZ=-1THENPRINT"You haven't got a ";C2$:GOTO750
1760IFOBZ(OBZ)<>-1THENPRINT"I'm not holding a ";C2$:GOTO750
1770OBZ(OBZ)=RN
1780IFOBZ<>23THEN1830
1790I=0
1800IF?FNJO(RN,I)=11THEN?FNJO(RN,I)=10:OBZ(23)=INT(RND(ABS(RN-1))):PRINT"There is a bright flash & the gray patch crumbles to reveal a hole in the wall.":GOTO490
1810IFI<5THENI=I+1:GOTO18^^
1820GOTO750
1830IFOBZ=0OROBZ=1OROBZ=3THENPRINT"You were lucky it didn't break."ELSE1850
1840IFOBZ=0THENPRINT"But the lamp has gone out." :LI=0:GOTO490 ELSE490
1850IFOBZ=20OROBZ=9THENPRINT"The ";OB$(OBZ);" fell in the middle of the floor & is making a funny noise.":EX=-1:GOTO490
1860IFOBZ=5ANDOBZ(11)=RN THENPRINT"You got the dragon right through the heart.":PRINT"The dead dragon turned black & is glowing hotter & hotter. It is now starting to smoke.":OBZ(11)=100:EX=-1:OB$(5)="White hot pile":GOTO490
1870IFOBZ=5THENPRINT"You didn't hit whatever you were aiming at. You're not very good with that are you?":GOTO750
1880IFOBZ=7ANDOBZ(12)=RN THENPRINT"As soon as the statue left your hand it came alive & doubled in size. The lizard ran like hell past you. The bat then turned back into a statue.":OBZ(12)=RND(49):GOTO490
1890GOTO1650
1900
1910REM * * * SHOOT routine
1920
1930IFNOT(OBZ(1)=-1OROBZ(2)=-1)THENPRINT"With what I haven't got a gun.":GOTO750
1940IFC2$=""THENPRINT"Shoot what ":GOTO750

```

```

1950IFNOTOBZ(1)=-1THEN2170
1960IFLA>5THENPRINT"Laser batteries are dead."ELSE1990
1970
1980IFOBZ(2)=-1THEN2170 ELSE750
1990IFOBZ<>-1THEN2140 ELSEGOSUB3960
2000IFOBZ=-1THENPRINT"Shoot what ?":GOTO750
2010FORI=0TO5
2020K9=?FNJO(RN,I)
2030IFK9=9ANDOBZ=7THEN?FNJO(RN,I)=8:PRINT"zzzzZAP ! The web melted away.":LA=LA+1:I=9:GOTO2070
2040IFK9=15ANDOBZ=6THEN?FNJO(RN,I)=14:PRINT"zzzzZAP! The laser blasted a hole in the steel panel. ";LA=LA+1:I=9:GOTO2070
2050IFK9=3ANDOBZ=0OROBZ=2ANDK9=7OROBZ=3ANDK9=11OROBZ=4ANDK9=13THEN2090
2060IFK9=5ANDOBZ=1THENPRINT"zzzzZAP BANG CRASH etc. The whole place collapses you are dead OK. Try again.":GOTO60
2070NEXT
2080IFI<9THENPRINT"Can't see one to shoot it.":GOTO750 ELSE490
2090LA=LA+1
2100PRINT"zzzzZAP! Had no effect."
2110I=9
2120GOTO2070
2130IFOBZ=1THENPRINT"You don't mean shoot at the laser ?":GOTO750
2140IFOBZ(OBZ)=-1THENPRINT"I'm not shooting it while I hold it.":GOTO750
2150IFOBZ(OBZ)<>RN THENPRINT"I can't see one to shoot it.":GOTO750
2160IFOBZ>9ANDOBZ<14OROBZ=16OROBZ=18THENPRINT"zzzzZAP! Didn't do much.":GOTO750 ELSEOBZ(OBZ)=1000:PRINT"zzzzZAP! Poof.The ";OB$(OBZ);" vanished in a flash & a cloud of smoke.":GOTO490
2170IFBU>5THENPRINT"Gun out of bullets.":GOTO750
2180IF OBZ<>-1THEN2260
2190FORI=0TO6
2200IFINSTR(WA$(I),C2$)THENOBZ=I
2210 NEXT

```

```

2220 IFOBZ=-1THENPRINT"Shoot at a what ?":GOTO75
0
2230PRINT"Bang ! That was a waste of a bullet"
2240BU=BU+1
2250GOTO750
2260IFOBZ(OBZ)<>RNTHENPRINT"This is a gun not a
guided boomerang.":GOTO750 ELSEIFNOT(OBZ=15OROBZ
=17)THENPRINT"Bang ! Just bounced off.":IFRND(8)=
8THENPRINT"But then hit you. You are now DEAD.
    Try again.":GOTO60 ELSE750
2270PRINT"That got the ";OB$(OBZ);" it's crawled
away."
2280BU=BU+1
2290OBZ(OBZ)=RND(49)
2300GOTO750
2310
2320REM * * * LIGHT routine
2330
2340IFC$=""THENPRINT"Light what ?":GOTO750
2350IFOBZ<>0THENPRINT"Sorry I only know how to l
ight a lamp.":GOTO750
2360IFOBZ(0)<>-1THENPRINT"But you have not got a
lamp.":GOTO750
2370LI=-1
2380PRINT"Lamp on."
2390GOTO520
2400REM * * * WAVE routine
2410IFC2$=""THENPRINT"Ok. No one waved back.":GO
TOTO750
2420IFOBZ=-1THENPRINT"Wave a ";C2$;".":GOTO750
2430IFOBZ<>23THENPRINT"Ok. Nothing happened.":GO
TOTO750
2440I=0
2450K9=?FNJO(RN,I)
2460IFK9=17THEN`=FNJO(RN,I):?`=?? EOR1:PRINT"The
re is a loud rumbling sound & the whole place sha
kes.":GOTO490
2470I=I+1
2480IFI=6THENOBZ=-2:GOTO2430 ELSE2450
2490
2500REM * * * RUB routine

```



```

2510
2520IFC2$=""THENPRINT"Rub what?":GOTO750
2530IFOBX=-1THENPRINT"No, don't be silly":GOTO75
0
2540IFOBX<>23THENPRINT"You're not getting me doi
ng that.":GOTO750
2550I=0
2560K9=?FNJO(RN,I)
2570IFK9=5THENPRINT"There is a sliding sound & t
he sides of the pillar slide away leaving a spira
l stair case.":`=FNJO(RN,I):?`=?? EOR1:GOTO490
2580I=I+1
2590IFI=6THENPRINT"Well nothing happened & I fee
l silly doing this.":GOTO750 ELSE2560
2600
2610REM * * * HELP routine
2620
2630IFEX=0THEN2670
2640PRINT"Don't stand there something's going to
go BANG. Quick do something.";
2650LINEINPUT": "C$
2660IFC$=""THEN750 ELSE800
2670IFRN=49THENPRINT"I think that sign says 'DAN
GER Falling Rocks.'
2680PRINT"You are ";50-RN;" caves away from the
exit if you don't take a short cut."
2690IF00=0THENPRINT"Every room has at least two
ways in & out.":IFOBX(23)<>-1THENPRINT"The dark
rod of Muton is very good at opening some of t
hem."
2700IF00<>0THENPRINT"Its DANGEROUS to stand that
close to the ";OB$(00);"."
2710IFRN<>0THENPRINT"If you want to go back to t
he last room then go ";DI$(BD);"."
2720GOTO490
2730
2740REM * * * GO routine
2750
2760IFOBX<>-1THENPRINT"I do not recognize ";C2$;
" as a direction":GOTO750
2770IFC2$=""THENPRINT"Don't get excited but which

```

```

way ?":GOTO750
2780I=0
2790IFINSTR(DI$(I),C2$)THENDI=I:GOTO2820
2800I=I+1
2810IFI>5THENDI=-1:ELSE2790
2820IFDI=-1THENOBZ=0:GOTO2760
2830REM Move in dirrection DI
2840DO=?FNJO(RN,DI)
2850IFDO=0ORNOT(2*INT(DO/2)=DO)THENPRINTCHR$129"
Can't go that way":GOTO750
2860DO=DO-2
2870IFDO ANDDI<>BD THENPRINT"The ";OB$(DO);" is
blocking your way.":GOTO750
2880LR=RN
2890RN=?FNJO(RN,DI)+1)
2900DI=FNOP(DI)
2910J=?FNJO(RN,DI)
2920J=2*INT(J/2)
2930?FNJO(RN,DI)=J
2940BD=DI
2950TH=0
2960IFEX=0THEN3040
2970EX=0
2980FORI=0TO30
2990IFOBZ(I)=LR THENOBZ(I)=100
3000NEXTI
3010PRINTCHR$130"There was a loud bang behind yo
u."
3020FORI=0TO1000
3030NEXT
3040IFPOS(0)<>0THENPRINT
3050PRINTSTRING$(40,"~");
3060GOTO520
3070
3080REM * * * OPEN routine
3090
3100IFC2$=""THENPRINT"Open what?":GOTO750
3110IFOBZ<>-1THENPRINT"I can't open a ";OB$(OBZ)
;".":GOTO750
3120GOSUB3960
3130IFOBZ<>4THENPRINT"Open a ";C2$;"?":GOTO750
3140I=0

```

```

3150K9=?FNJO(RN,I)
3160IFK9<>13THEN3210
3170`=FNJO(RN,I)
3180?`=??` EOR1
3190PRINT"Ok."
3200IFRND(4)=1THENPRINT"Behind the door there wa
s a monster of a spider that just wrapped you up
for its lunch.      Hard luck try again":GOTO60
ELSE520
3210 I=I+1
3220IFI=6THENPRINT"What door?":GOTO750
3230GOTO3150
3240
3250REM * * * JUMP routine
3260
3270PRINT"I'm not jumping in these shoes."
3280GOTO490
3290
3300REM * * * INVENTORY routine
3310
3320PRINT"You are carrying :"
3330FORI=0TO30
3340IFOBZ(I)=-1THENPRINTOB$(I),;
3350NEXT
3360GOTO520
3370
3380REM * * * PUSH routine
3390
3400IFC$=""THENPRINT"Push what ?":GOTO750
3410IFOBZ<>-1THEN3520
3420GDSUB3960
3430IFOBZ=-1THENPRINT"Push a ";C$;" ?":GOTO750
3440I=0
3450K9=?FNJO(RN,I)
3460IFK9=7ANDOBZ=2THENPRINT"There is a rumbling
sound & the blocks slide forward to make steps go
ing up to the hole.";:`=FNJO(RN,I):?`=??` EOR1:GOT
O490
3470IFK9=3ANDOBZ=0THENPRINT"It crumbles away sho
wing a hole.";:`=FNJO(RN,I):?`=??` EOR1:GOTO490
3480IFK9=9ANDOBZ=7THENPRINT"You're stuck to the

```

```

web.":INPUTLINE": "A$:FORI=0TO1000:NEXT:PRINT"The
re is a giant spider coming at you. Well you did
n't last long after that. Try again.":GOTO60
3490IFI<5THENI=I+1:GOTO3450
3500PRINT"Did not move at all."
3510GOTO750
3520IFOBZ(OBZ)<>RN THENPRINT"Push what ";C2$;".
:GOTO750
3530IFOBZ<10THENPRINT"OK.":GOTO750
3540IFOBZ<18OROBZ=27THEN1190
3550IFOBZ<24OROBZ>27THENPRINT"OK. Why not?":GOTO
750
3560PRINT"How ?"
3570GOTO750
3580
3590REM * * * U routine
3600
3610DI=0
3620GOTO2830
3630
3640REM * * * D routine
3650
3660DI=1
3670GOTO2830
3680
3690 REM * * * N routine
3700
3710DI=2
3720GOTO2830
3730
3740REM * * * S routine
3750
3760DI=3
3770GOTO2830
3780
3790REM * * * E routine
3800
3810DI=4
3820GOTO2830
3830
3840REM * * * W routine
3850

```

```

3860DI=5
3870GOTO2830
3880
3890REM * * * QUIT routine
3900
3910INPUT"You give up",A$
3920IFLEFT$(A$,1)="Y"THENPRINT"Disgusting, a st
ar cruiser crew member would never give up.":FORI
=0TO1000:NEXT:END ELSE490
3930
3940REM See if C2$ is a key word for a exit.
3950
3960FORI=0TO7
3970IFINSTR(WA$(I),C2$)THENOBZ=I
3980NEXTI
3990RETURN
4000END
4010
4020
4030
4040
4050REM Initalise
4060FORI=0TO21
4070READC$(I)
4080NEXT
4090FORI=0TO30
4100READOB$(I)
4110NEXT
4120RESTORE5110
4130FORI=0TO15
4140READDO$(I)
4150NEXT
4160FORI=0TO7
4170READWA$(I)
4180NEXT
4190FORI=0TO5
4200READDI$(I)
4210NEXT
4220
4230REM Randomly join rooms
4240

```

```

4250 REM Columns of JO( ) U D N S E W ( , ( 0 - 5
))
4260
4270 REM Link rooms in order
4280
4290 FOR I=0 TO 49
4300 NR=1
4310 GOSUB 4350
4320 NEXT
4330 GOTO 4550
4340
4350 DI=RND(6)-1
4360 DE=FNJO(DI)
4370 IF ?FNJO(I,DI)=0 AND ?FNJO(I,DE)=0 THEN 4410 ELSE 4
350
4380 DI=RND(6)-1
4390 DE=FNJO(DI)
4400 IF ?FNJO(I,DI)<>0 OR ?FNJO(I+NR,DE)<>0 THEN RETUR
N
4410 IF DI=1 THEN DO=RND(4)-1:GOTO 4440
4420 IF DI=0 THEN DO=RND(2)+3:GOTO 4440
4430 DO=RND(10)+5
4440 ?FNJO(I,DI)=DO+2
4450 ?(FNJO(I,DI)+1)=I+NR
4460 ?(FNJO(I+NR,DE)+1)=I
4470 IF DE=1 THEN DO=RND(4)-1:GOTO 4500
4480 IF DE=0 THEN DO=RND(2)+3:GOTO 4500
4490 DO=RND(10)+5
4500 ?FNJO(I+NR,DE)=DO+2
4510 RETURN
4520
4530 REM Make extra links between rooms at random
4540
4550 FOR I=1 TO 46
4560 NR=RND(47)-1
4570 GOSUB 4380
4580 NEXT I
4590
4600 REM Place objects in rooms
4610
4620 FOR I=2 TO 30

```

```

46300BX(1)=RND(50)
4640NEXT
46500BX(0)=-1
46600BX(1)=-1
46700BX(23)=0
4680EX=FALSE
4690LI=FALSE
4700RN=0
471000=0
4720LA=0
4730BD=-1
4740RETURN
4750
4760REM Start of data
4770
4780REM Verbs
4790
4800DATA GET,TAKE,DROP,THROW,SHOOT,LIGHT,WAVE,RU
B,HELP,GO,OPEN,JUMP,INVENTORY,PUSH,U,D,N,S,E,W,QU
IT,LOOK
4810
4820REM Objects
4830
4840 DATA LAMP,LASER PISTOL,GUN,AIR BOTTLE,BATTE
RY,SPEAR,STONE,STATUE OF A BAT
4850DATASHIELD,GRENAD,Boulder,DRAGON,GIANT LIZA
RD,LARGE CARNIVOROUS TREE,BRIGHT RED STONE,MUTANT
ASTRONAUT,GRIFFIN,MUTANT MOUSE,ROSE BUSH
4860DATA GREEN WOODPECKER,YELLOW STONE,LUMP OF M
EAT,WATER BOTTLE,BLACK STICK,POOL OF WATER,POOL O
F LIQUID,WATER,RED STICK,GREEN FLY,MOLE,SMALL BLA
CK BOX
4870
4880REM Room discriptions
4890
4900 DATA at the bottom of very deep & dark pit
.You can only just see the green sky with red clo
uds above,in a narrow winding tunnel
4910DATA in small low roofed cave,surrounded by
a forest of towering grass in a very large cave
4920DATA in a cave with green & orange slime on
the walls,walking inbetween swamps & quick sand,i

```

n a square room with walls made of a strange metal

4930 DATA under a large rock hanging from the ceiling of a large cave by a dead vine, in plane room

4930 DATA under a large rock hanging from the ceiling of a large cave by a dead vine, in plane room

4940 DATA standing under a huge web with a smelly dead monster of a spider beside you

4950 DATA on the massive roots of some tree in the cave above, in a chalk cave surrounded by the empty & torn spacesuits of some of the other members of crew

4960 DATA in a very small cave, in a very large cave, in a long straight tunnel, between two large stalactites, in a jungle of dead plants

4970 DATA in a cave half filled with sand, in clouds of yellow smoke in between mini-valcanoes & lava flows

4980 DATA in a cavern with the floor covered with shallow pits, balancing on the strands of a large web, up to your knees in dead rats p.s. IT SMELLS

4990 DATA inside the skeleton of some huge animal that must have got jammed in this cave

5000 DATA in a large hall lined with statues of weird creatures, in a room with straw on the floor with a strong smell of animals, at the middle of a large room with the ceiling too high to see

5010 DATA in a blue corridor, in a blue corridor, in a blue corridor, in a red corridor, in a red corridor, in a yellow chamber, in a jungle of dead plants, bent over in a very large cave but only 4 ft high

5020 DATA in a blue corridor, squeezed into a very small cave, up to your chest in mud, getting very wet there is green slimy water falling from the roof of this cave, in a blue corridor

5030 DATA in a large cave containing the ruins of three temple like buildings, surrounded by a forest of towering grass in a very large cave, on narrow pathways on the sides of deep pits, squeezing between huge boulders that fill this cave

5040 DATA in a round room containing broken elect

ronic equipment, in a room containing a large computer there are video cameras following you as you move, in a small rocky corridor, in a room with walls of quartz

5050 DATA in a room with paintings of monsters on the walls, in a small clean room, in a room in which part of the ceiling has collapsed on the floor

5060 DATA a room containing piles of old equipment & a sign on the wall that says 'ghhj njjj hjh sa dj bjkkhg vgjh'

5070 DATA in the doorway of an old 'P' class one man space shuttle with a full load of fuel & supplies

5080

5090 REM Exit descriptions

5100

5110 DATA a hole in the floor with steps leading down, a stone slab on the floor, spiral stair case going down, a large metal pillar sticking from the floor

5120 DATA some steps in the wall that go up through a hole, some blocks in a regular pattern on the wall below a hole

5130 DATA an archway, a giant web blocking an archway, a hole in the wall, a gray patch on the wall, an open door, a closed black door

5140 DATA a large hole in a steel panel, a steel panel, a large crack in the wall, a small crack in the wall

5150

5160 REM Key words for exits

5170

5180 DATA STONE SLAB, PILLAR, BLOCKS, GRAY PATCH, BLACK DOOR, SMALL CRACK, STEEL PANEL, GIANT WEB

5190

5200 DATA UP, DOWN, NORTH, SOUTH, EAST, WEST

5210

5220 REM Convert number between odd & even

5230

5240 DEFFNOP(X)=X EOR 1

5250

5260 REM Calc address next room pointer from room

```

I in dirrection DI
5270
5280DEFFNJO(I,DI)=JO+I*12+DI*2
5290
5300REM Find discription of room N
5310
5320DEFFNRO(N)
5330LOCALIZ,A$
5340RESTORE4900
5350FORIZ=0TON
5360READA$
5370NEXT
5380=A$
5390
5400REM INSTR Avoids basic 1 bug. (eg. Crash if
length search string greater then length of main
string.)
5410
5420DEFFNINSTR(M$,S$)
5430IFLENM$<LENS$THEN=0 ELSE=INSTR(M$,S$)
5440
5450REM Format text in teletext colour COL
5460
5470DEFPROC(TEXT$,COL)
5480LOCALI,J,L,K
5490$TBUFFZ=TEXT$
5500I=TBUFFZ-1
5510VDUCOL
5520REPEATJ=I
5530REPEATJ=J+1
5540UNTIL?J=13OR?J=32
5550IFPOS+J-I>40THENVDU13,10
5560IFPOS=0THENVDUCOL
5570FORK=I+1TOJ
5580VDU-?K*(?K<>13)
5590NEXT
5600I=J
5610UNTIL?J=13
5620ENDPROC

```

ON THE WAY TO THE INTERVIEW

You are at the tube station and have to get to a job interview at Slave Driver International and you must not be late. On the way there are some small problems to solve and dangers to be cautious of like the number 9 bus, etc. The adventure game is text based but does show a map of the area, streets and buildings, around your position.

This program was originally written for a PET and therefore does not make full use of procedures and functions. The game accepts the following verbs as part of commands "GO". "LOOK", "QUIT", "GET", "DROP", "ASK", "CASH" & "SHOOT". Commands like "GET SHOPPING BAG" or just "GET BAG" will be understood. The directions "NORTH", "SOUTH", "EAST", "WEST", "IN" and "OUT" are understood when used after the word "GO".

When you are playing this game remember that it is against the clock as you must not be late for the interview.

```

10REM ON THE WAY TO THE INTERVIEW
20MODE4
30VDU23,224,48,48,16,48,80,40,72,8
40VDU28,0,31,39,24
50VDU24,0;260;1279;1023;
60PRINT"      ON THE WAY TO THE INTERVIEW"
70DEFFNPO(D)=XP*100+YP
80MS=10
90NS=31
100LX=3
110LY=4
120REM Store start time
130T0=TIME
140NB=6
150NV=5
160XP=LX
170YP=LY
180DIM MAP(MS,MS),ST$(NS),DI$(5),OB(NB),OB$(NB)
,V$(NV)
190REM Dimension the street map,street names,di
rrections,object positions,object names,verbs
200
210REM INIT
220
230GOSUB1040
240
250REM-----MAIN LOOP
260
270GOSUB700
280
290REM LOOK
300
310GOSUB1550
320
330REM INPUT
340IFC1$="GO"THENGOTO1680
350IFC1$="LOOK"THEN450
360IFC1$="QUIT"THENPRINT"I WOULD GO & PLAY SNAP
":END
370IFLEFT$(C1$,3)="INV"THEN2300
380GOTO2620
390

```

```

400REM VERBS + OBJECTS
410
420IFC2$=""THENPRINTC1$;" ?":GOTO310
430PRINT"I DO NOT KNOW HOW TO "C1$;" SOMETHING.
"
440
450REM VERBS FINISH HERE
460
470IFFNPO(0)=1000THEN560
480IFSN=1049 ANDOB(4)=-1THENPRINT"YOU HAVE JUST
BEEN ARRESTED FOR          ATTEMPTED ARMED ROBBER
Y.":GOTO650
490IFSN=1050THENPRINT"YOU JUST HAD A RIDE IN A
POLICE CAR TO  THE POLICE STATION.":XP=0:YF=0:IFOB
B(4)=-1THENPRINT"THE GUN NEEDED A BIT OF HIDING."
500GOSUB3450
510REM NASTY RANDOM EVENTS
520GOTO270
530
540REM-----REPEAT MAIN LOOP
550
560T=(TIME-T0)/6000
570PRINT"AND THERE AT THE END OF THE ROAD IS"
580PRINT"SLAVE DRIVER INTERNATIONAL."
590PRINT"WELL YOU'RE THERE."
600IFT<7THENPRINT"YOU MADE IT TO THE INTERVIEW
ON TIME.":END
610IFT<10THENPRINT"BUT TOO LATE.":END
620PRINT"BUT THEY HAVE ALL GONE HOME FOR THE NI
GHT."
630END
640REM ---- BYE
650PRINT"WELL I DON'T THINK YOU ARE GOING TO GE
T TO THE INTERVIEW NOW."
660PRINT"I AM GIVING UP ON YOU AFTER ";INT((TIM
E-T0)/6000);" MINUTES OF MY VALUABLE TIME."
670END
680
690
700REM LOOK ROUTINE
710
720

```

```

730REM Do graphics
740
750PROCSHOW(XP,YP)
760
770REM Do text
780
790F=0
800HE=MAP(XP,YP)
810IFHE AND 1024THENPRINT"YOU ARE INSIDE ";ST$(
HE AND1023);" ";:GOTO890
820PRINT"YOU ARE ON ";ST$(HE);" ";
830FORI=0TO3
840C2$=DI$(I)
850GOSUB2140
860IFSN=HE AND F=0THENPRINT"WHICH GOES ";:F=1
870IFSN=HE THENPRINT" ";DI$(I);
880NEXT
890PRINT"."
900FORI=0TO3
910C2$=DI$(I)
920GOSUB2140
930IFNOTND AND SN<>HE THENPRINT"TO THE ";DI$(I)
;" IS ";ST$(SN AND1023)".
940NEXT
950F=0
960P=FNPO(0)
970FORI=1TONB
980IFOB(I)<>P THEN1010
990IFI<3THENPRINT"STANDING ON THE CORNER IS A "
;OB$(I);"." :GOTO1010
1000PRINT"ON THE GROUND IS A ";OB$(I);"."
1010NEXT
1020RETURN
1030
1040REM SET UP MAP ,DIRRECTIONS ETC
1050DI$(0)="NORTH"
1060DI$(1)="SOUTH"
1070DI$(2)="EAST"
1080DI$(3)="WEST"
1090SN=1
1100SH=0

```

```

1110RESTORE
1120READS$
1130GOSUB1370
1140
1150REM READ STREET
1160
1170READS$
1180IFS$<>"*ANDS$<>"$ "THEN1130
1190IFS$=" $"THEN1260
1200REM ### SHOP ETC. DATA
1210SH=1
1220GOTO1120
1230
1240REM READ OBJECTS
1250
1260FORI=1TONB
1270READOB$(I),OB(I)
1280NEXT
1290
1300REM READ NEW VERBS
1310
1320FORI=1TONV
1330READV$(I)
1340NEXT
1350RETURN
1360
1370ST$(SN)=S$
1380READX,Y
1390IFX<0OR Y<0THENSN=SN+1:RETURN
1400MAP(X,Y)=SN OR SH*1024
1410GOTO1380
1420
1430P=0
1440FORI=1TO LEN(L$)
1450IFMID$(L$,I,1)=" "THENP=I:I=3000
1460NEXT
1470RETURN
1480
1490IFJ$="" OR LEFT$(J$,1)<>" "THENRETURN
1500J$=MID$(J$,2)
1510GOTO1490

```

```

1520
1530REM Ask for command
1540
1550INPUT"}"L$
1560CLG
1570IFL$=""THENPRINT"SILENT TYPE AH.":GOTO1550
1580GOSUB1990
1590SE$=" "
1600GOSUB1430
1610IFP=0THENC1$=L$:C2$="":RETURN
1620C1$=LEFT$(L$,P-1)
1630J$=MID$(L$,P+1)
1640GOSUB1490
1650C2$=J$
1660RETURN
1670
1680IFC2$=""THENPRINT"OK BUT WHICH WAY?":GOTO450
1690IFC2$="IN"THEN1810
1700IFC2$="OUT"THEN1910
1710GOSUB2140
1720IFND THENPRINT"YOU CAN'T GO THAT WAY.":GOTO4
50
1730REM **** CODE FOR TESTING FOR OBSTRUCTIONS
1740GOSUB2430
1750IF NOT ND THENLX=XP:XP=X:LY=YP:YP=Y:GOTO450
1760GOTO450
1770
1780
1790REM GO IN
1800
1810FORI=0TO3
1820C2$=DI$(I)
1830GOSUB2140
1840IFSN AND 1024THENI=3000
1850NEXT
1860IFI<3000THENPRINT"IN WHAT ?":GOTO450
1870GOTO1720
1880
1890REM GO OUT
1900
1910IF0=(HE AND1024)THENPRINT"YOU ARE NOT IN.":G

```



```

0T0450
1920FORI=0T03
1930C2$=DI$(I)
1940GOSUB2140
1950IF0=(SN AND 1024)ANDSN THENI=3000
1960NEXT
1970GOTO1720
1980
19900$=""
2000
2010REM REMOVE EFFECTS OF SHIFT KEY
2020
2030FORI=1 TO LEN(L$)
2040Z=ASC(MID$(L$,I,1))
2050Z=Z AND 127
20600$=0$+CHR$(Z)
2070NEXT
2080L$=0$
2090RETURN
2100
2110
2120REM READ MAP AROUND XP,YP
2130
2140X=XP
2150Y=YP
2160SN=0
2170C3$=LEFT$(C2$,1)
2180IFC3$="N"THENY=Y-1
2190IFC3$="S"THENY=Y+1
2200IFC3$="E"THENX=X+1
2210IFC3$="W"THENX=X-1
2220IFX+Y=XP+YP THENPRINT"I WISH YOU WOULD GO ";
C2$ !"
2230NO=X>10 OR X<0 OR Y<0 OR Y>10
2240IFNOTNO THENNO=NO OR MAP(X,Y)=0
2250IFNOTNO THEN SN=MAP(X,Y)
2260RETURN
2270
2280REM INVENTORY
2290
2300F=0
2310PRINT"YOU ARE HOLDING :";

```

```

2320FORI=1TONB
2330IFOB(I)<>-1THEN2360
2340F=F OR 3454
2350PRINT " ";OB$(I);
2360NEXT
2370IFF=0THEN PRINT"NOT A LOT";
2380PRINT"."
2390GOTO450
2400
2410REM OBSTRUCTIONS
2420
2430IFSN<>13THENRETURN
2440IFOB(6)=-1AND HE<>13THENPRINT"IT COST YOU A
BIT TO GET IN HERE.":RETURN
2450IFOB(6)=-1THENRETURN
2460ND=-1
2470PRINT"SORRY IT COSTS MONEY TO ENTER THE PARK
"
2480IFOB(5)=-1THENPRINT"AND WE DO NOT EXCEPT CHE
QUES."
2490RETURN
2500
2510REM PEOPLE TROUBLE
2520
2530IFOB(4)=-1THENPRINT"THE TRAMP JUST RAN AWAY.
":OB(2)=-2
2540PRINT"YOU JUST HAD A BAD EXPIENCE WITH A TRA
MP."
2550GOTO650
2560IFOB(4)<>-1THENPRINT"WHY SHOULD SHE KNOW.":G
OTO450
2570PRINT"SHE JUST LOCKED YOU UP FOR BEING IN PO
SSSESSION OF A DEADLY WEAPON."
2580GOTO650
2590
2600REM VERBS RELATED TO OBJECTS
2610
2620IFC2$=""THEN420
2630
2640REM FIND OBJECT NUMBER
2650

```

```

2660IT=0
2670FORJ=1 TONB
2680C$=OB$(J)
2690GOSUB3290
2700IFF0 THEN IT=J:J=3000
2710NEXT
2720IFIT=0THENPRINT"WHATS A ";C2$;"?":GOTO450
2730V=0
2740
2750REM FIND VERB NUMBER
2760
2770FORI=1TONV
2780IFC1$=V$(I)THENV=I:I=3000
2790NEXT
2800IFV=0THENPRINT"NO IT'S AGAINST MY UNION RULE
S.":GOTO450
2810ON V GOTO 2910,3010,3070,3150,3240
2820
2830REM VERB ROUTINES
2840
2850GOTO420
2860
2870REM NOT KNOW
2880
2890REM *GET*
2900
2910IFOB(IT)=-1THENPRINT"YOU ARE ALREADY HOLDING
IT.":GOTO450
2920IFOB(IT)<>FNPD(0)THENPRINT"I CAN'T SEE A ";O
B$(IT);".":GOTO450
2930IFIT=2 THEN2530
2940IFIT=1THENPRINT"NO. I THINK SHE IS A BIT OLD
FOR YOU.":GOTO450
2950IFIT=3THENPRINT" B A N G ! PARCEL BOMB AH!
":GOTO650
2960OB(IT)=-1
2970GOTO450
2980 .
2990REM *DROP*
3000
3010IFOB(IT)<>-1THENPRINT"YOU ARE NOT HOLDING A

```

```

";OB$(IT);".":GOTO450
3020OB(IT)=FNPO(0)
3030GOTO450
3040
3050REM *ASK*
3060
3070IFOB(IT)<>FNPO(0)THENPRINT"WHAT ";OB$(IT);"?
":GOTO450
3080IFIT=2THEN2530
3090IFIT=1THEN2560
3100PRINT"DO YOU ALWAYS TALK TO ";OB$(IT);"?
3110GOTO450
3120
3130REM *CASH*
3140
3150IFIT<>5THENPRINT"I'LL CASH YOU IN A MINUTE."
:GOTO450
3160IFFNPO(0)<>308THENPRINT"I THINK YOU NEED A B
ANK.":GOTO450
3170OB(5)=-2
3180OB(6)=-1
3190PRINT"OK. BUT THE CHEQUE WAS NOT FOR MUCH."
3200GOTO450
3210
3220REM *SHOOT*
3230
3240IFOB(4)<>-1THENPRINT"I CAN'T SHOOT ANYTHING
WITH MY FINGER.":GOTO450
3250IFOB(IT)<>FNPO(0)THENPRINT"SORRY I CAN'T SEE
ONE.":GOTO450
3260PRINT"I'M NOT HAVING ANYTHING TO DO WITH YOU
SHOOTING THINGS."
3270GOTO650
3280
3290FO=-1
3300
3310REM SEARCH FOR WORDS IN A STRING
3320
3330IFLEFT$(C$,LEN(C2$))=C2$THENRETURN
3340L$=C$
3350SE$=" "
3360GOSUB1430

```

```

3370IFP=0THENFO=0:RETURN
3380L$=MID$(L$,P+1)
3390IFLEFT$(L$,LEN(C2$))=C2$THENRETURN
3400GOSUB1430
3410GOTO3370
3420
3430REM RANDOM EVENTS
3440
3450IFRND(1)>.047THENRETURN
3460IFOB(4)=-1THENPRINT"BANG THAT MADE YOUR EYE
S WATER."/"YOU SHOULD NOT HAVE PUT THAT GUN IN TH
E FRONT OF YOUR BELT.":GOTO1860
3470IFHE AND1024THEN3510
3480IFRND(1)>.8THENPRINT"SPLAT ! YOU ARE UNDER A
NUMBER 9 BUS":GOTO650
3490PRINT"THAT NUMBER ";INT(RND(1)*404);" BUS NE
ARLY HIT YOU."
3500RETURN
3510PRINT"THEY HAVE ALL GONE HOME & LOCKED YOU I
N."
3520GOTO650
3530
3540DEFPROC$SHOW(X,Y) LOCALIZ,JZ
3550CLG
3560FORIZ=-1TO1
3570FORJZ=-1TO1
3580IFFNMAP(X+JZ,Y+IZ)THENPROC$SHADE(JZ,IZ)
3590IFFNMAP(X+JZ,Y+IZ)AND1024THENPROC$DENSE(JZ,IZ
)
3600NEXT
3610NEXT
3620MOVE632,708
3630VDU5,224,4
3640ENDPROC
3650
3660DEFPROC$SHADE(X,Y) LOCALI
3670FORI=-100TO100STEP8
3680MOVE540+200*X,700-200*Y+I
3690DRAW740+200*X,700-200*Y+I
3700NEXT
3710ENDPROC

```

```

3720
3730DEFPROC DENSE(X,Y) LOCAL I
3740FOR I=-80 TO 80 STEP 4
3750MOVE 560+200*X,700-200*Y+I
3760DRAW 720+200*X,700-200*Y+I
3770NEXT
3780ENDPROC
3790
3800DEFFN MAP(X,Y)
3810IF X>MS OR X<0 THEN=0
3820IF Y>MS OR Y<0 THEN=0
3830=MAP(X,Y)
3840
3850
3860REM ROAD POSITION DATA
3870
3880DATA ROUND STREET,0,1,1,1,2,1,0,2,0,3
3890DATA 1,3,2,3,2,2,-1,-1
3900DATA END STREET,2,0,-1,-1
3910DATA FRED'S LANE,3,1,-1,-1
3920DATA STATION ROAD,4,0,4,1,4,2,4,3,4,4,4,5
3930DATA 4,6,4,7,-1,-1
3940DATA END ROAD,1,4,1,5,-1,-1
3950DATA CROOK STREET,0,5,0,6,0,7,1,7,1,8
3960DATA-1,-1
3970DATA THE WAY,2,6,3,6,-1,-1
3980DATA PRINCE ROAD,5,4,-1,-1
3990DATA SHORT CUT CLOSE ,6,4,6,5,6,6,-1,-1
4000DATA RETURN ROAD,5,6,-1,-1
4010DATA HOME STREET,0,9,1,9,2,9,3,9,4,9,6,9
4020DATA 5,9,-1,-1
4030DATA LATE STREET,6,8,7,8,-1,-1
4040DATA A PRIVATE PARK,8,5,8,6,8,7,8,8,9,6,9,7,
-1,-1
4050DATA SOMEONE'S DRIVE WAY,9,9,-1,-1
4060DATA INNER VIEW LANE,10,7,10,8,10,9,-1,-1
4070DATA COMMODORE STREET,8,4,9,4,10,4,-1,-1
4080DATA SHORT STREET,9,3,-1,-1
4090DATA LOST MANS ROAD,6,2,7,2,8,2,9,2,-1,-1
4100DATA STATION ROAD,9,1,10,1,-1,-1
4110DATA THE GOOD WAY,6,1,-1,-1
4120DATA THE PLACE,6,0,7,0,8,0,-1,-1

```

4130DATA DEAD MANS DRIVE,10,0,-1,-1
4140DATA "*"
4150
4160REM PLACE DATA
4170
4180DATA A POLICE STATION,0,0,-1,-1
4190DATA A TUBE STATION,3,4,-1,-1
4200DATA A BANK,3,8,-1,-1
4210DATA A PRIVATE HOUSE,1,10,-1,-1
4220DATA A DARK TUNNEL UNDER THE RAILWAY
4230DATA 6,10,7,10,8,10,9,10,-1,-1
4240DATA S D INTERNATIONAL,7,3,-1,-1
4250DATA FRED'S TEA SHOP,10,5,-1,-1
4260DATA "\$"
4270
4280REM OBJECT DATA
4290
4300DATA POLICE WOMAN,509
4310DATA TRAMP,103
4320DATA SHOPPING BAG,402
4330DATA GUN,7
4340DATA CHEQUE,-1
4350DATA MONEY,-2
4360
4370REM VERBS
4380
4390DATA GET,DROP,ASK,CASH
4400DATA SHOOT

DATA PROGRAM

DESCRIPTION

This program is a computer address book or diary. It can store a large number of items and each item can be up to nine lines long, but must be less than 250 characters long in total. You may store data either under a number between 1 and 1000 or under a date. For example:

5 June 1984

Have a bath.

After typing "RUN", the program displays a menu of the following choices:

UPDATE/DISPLAY: will allow you to change record of a given data or number.

FIND: being a search routine, this will display all records containing the characters or words you are looking for, until you find the record you want to look at or change. At each record, it will give you the option to change it.

REMOVE: this will remove a record of a given number or date.

LIST ALL: shows all the records currently in the program.

SAVE: allows you to save the program and data to the current filing system eg. tape or disk.

END: this terminates the program, after giving you an option to save it first.

When using the update/display and find commands, the program will display the record and ask "CHANGE". If you reply "Y" and you have a new version of the operating system, then the program will move the copy cursor to each line of the old record so you can type the new one in. You can then use the copy key, without first moving the cursor yourself. If you just press return, then the old line will not be changed.

All actions that involve changes to the stored records have the options to make the change or to abort, using the keys F0 and F1.

PROGRAM STRUCTURE

The program starts with initialising variables, arrays and reading in the full names of the months from data statements. Next, procedure MENU is entered, this prints up the choices available. When a key is pressed the appropriate procedure is called. If the "E" is pressed then the program goes on to procedure FINISH, which gives the user an option to save the data and program before ending.

FUNCTIONS AND PROCEDURES

FNLINE: This returns as its value the line number of the current data statement.

If the start of the current data statement can not be found, then the value zero is returned. It uses the 16 bit pointer to the end of the

current data item at address HEX IC. It then works back from there until it finds a HEX DEC, the token for "DATA". It then assumes that the three bytes before that are the line number (two byte) and the length of line.

PROCOSCLI: This is already defined as "OSCLI" in the second version of the BBC basic so if you have this version, you do not need this definition (lines 56-66). You just need to replace each occurrence of "PROCOSCLI" with "OSCLI". This procedure takes a string as a parameter and passes it to the operating system command line interpreter. Eg. PROCOSCLI ("CAT") would have the same effect as typing *CAT in command code (outside a basic program).

PROCMENU: As described above.

FNENTRY: This is to take a number or date from the user, convert it to a line number and to repeat it, until it finds a valid line number. This is then returned as the value of the function.

FNDTOL: Takes a string containing the date in the form DDDMMYYYY and converts it into a program line number. Lines 124-130 calculate the number of the month. If it can not convert the line number because of an error in the parameter string, then line zero is returned.

PROCUP: Called directly from the menu, this displays and gives the option to update a stored record or to add new records. First it uses FNENTRY to get a line number of a record. Then it tries to set the pointer to the current record to a DATA statement at this line number, using RESTORE L, if this fails then the ON ERROR tells it to go to the input routine PROCINREC to ask if the blank record has to be changed. If a data statement was found with that line number, then it is displayed first.

FNLTO: This takes a line number and converts it to a record number or to a date and returns it in a string.

PROCCOPY: If you do not have a new version of the operating system, then this procedure and all reference to it will have to be left out of the program. It has the effect of pressing the grey up arrow key ten times. This is done by putting ten characters 255 in the keyboard buffer.

PROCINREC: If the question "CHANGE" is not answered with "Y", then it returns via a normal ENDPROC. Otherwise it takes nine lines of text from the keyboard and any blank lines are replaced with the value in the old record. The lines are then formatted as a basic DATA statement with the correct line number for the current record. Function key 0 is programmed using PROCOSCLI to hold a "#", followed by the new line of basic text, a return and a RUN return. When a "#" is typed then the program ends. If the "#" came from F0 or F1 then the new line of basic program is left in the keyboard buffer with a RUN command, so the new line is entered and the program restarts.

PROCLIST: This lists all records contained in the program, until the "Out of Data" error occurs. Then the program is run from the start via PROCRESTART.

PROCSTART: When a key is pressed, run the program from the beginning.

PROFIND: Takes a string of characters from the keyboard and uses INSTR to search each data item for it. If it finds a match, then it finds the line number of the current data statement and displays the record it contains. It then asks CHANGE etc. It repeats until all of the matches have been shown or a record is changed. PROCFINISH is only left via PROCSTART.

PROCREMOVE: Removes a data statement from program using the function keys as in PROCINREC.

PROCSAVE: Uses the operating system command line interpreter, to save the program and data statements using *SAVE.

PROCFINISH: Uses and gives an option to use PROCSAVE before doing an END.

```

5REM DATAPROG
6REM ENTRY IS LOCATED BY IT'S DATE OR IT'S EN
TRY NUMBER
7REM DATES TAKE THE FORM DDMMYYYY
8REM                      EG.   5SEP1983
10MODE7
12*FX18
14*K.1#RUNIM
16DIM COMLZ 256,M$(12),R$(9)
18Y0=1983
20RESTORE
22FORI=1TO12
24READM$(I)
26NEXT
28PROCENUE
30PROCFINISH
32END
34
36DEF FNLIN
38DPZ=!81C AND&FFFF
40IFDPZ=PAGE THEN=0
42REPEAT
44DPZ=DPZ-1
46UNTIL?DPZ=&DC
48DPZ=DPZ-4
50IFDPZ<PAGE THEN=0
52=DPZ?1*256+DPZ?2
54
56DEF PROCOSCLI(A$)
58XZ=COMLZ
60YZ=XZ DIV 256
62$COMLZ=A$
64CALL&FFF7
66ENDPROC
68
70DEFPROCENUE
72REPEAT
74PRINT"" U: UPDATE /DISPLAY"
76PRINT"   F: FIND"
78PRINT"   R: REMOVE"
80PRINT"   L: LIST ALL"

```

```

82PRINT"      S: SAVE PROG & DATA"
84PRINT"      E: END"
86C$=GET$
88IFC$="U"THENPROCUP
90IFC$="F"THENPROCFIND
92IFC$="R"THENPROCREMOVE
94IFC$="L"THENPROCLIST
96IFC$="S"THENPROCSAVE:RUN
98UNTILC$="E"
100ENDPROC
102
104DEF FNENTRY LOCAL A$,V
106REPEAT
108INPUTLINE"/" Entry date or number "A$
110IFLENA$<4THENA=VALA$:UNTILA>0 ANDA<1000:=A+1
000
112A=FNDTOL(A$)
114UNTILA<>0 ANDA<87FFF
116=A
118
120DEF FNDTOL(D$) LOCAL D,M$,P,Y
122D=VALD$
124M$=LEFT$(RIGHT$(D$,7),3)
126P=INSTR("JAN*FEB*MAR*APR*MAY*JUN*JUL*AUG*SE
P*OCT*NOV*DEC",M$)
128IFP=0THEN=0
130M=P DIV4+1
132Y=VALRIGHT$(D$,4)
133IFY<Y0 THEN=0
134=2000+(Y-Y0)*400+M*32+D
136
138DEF PROCUP
140L=FNENTRY
142CLS
144ONERROR ONERROR OFF:PROCINREC:PROCRESTART
146RESTORE L
148ONERROR OFF
150PROCDISPLAY
152PROCINREC
153PROCRESTART
154

```

```

156DEFPROCDISPLAY
158CLS
160PRINT"*** ";FNLTOB(L)
162FORI=1TO9
164READR$(I)
166PRINTTAB(7,I)":"R$(I)
168NEXT
170ENDPROC
172
174DEF FNLTOB(L) LOCAL Y,M,D
176IFL<2000THEN=STR$(L-1000)
178L=L-2000
180Y=L DIV400+Y0
182M=L MOD 400 DIV32
184D=L MOD 400 MOD32
186=STR$D+M$(M)+STR$Y
188
190DATAJANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JU
LY,AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
192DEF FNQUO(A$)
194IFA$=""THEN=""
196=""""+A$+""""
198
200DEF PROCCOPY LOCAL IX
202*FX15,1
204FORIX=1TO10
206*FX138,0,255
208NEXT
214ENDPROC
216
218DEF PROCINREC LOCALIX
220PRINTCHR$30"*** ";FNLTOB(L);" *** CHANGE :"
221IFGET$<>"Y"THENENDPROC
222FORIX=1TO9
224PRINTTAB(7,IX+10);
226PROCCOPY
228INPUTLINEA$
230IFA$<>" "THENR$(IX)=A$
232K$=K$+FNQUO(R$(IX))+","
234NEXT
236K$="K.0#"+STR$L+"DATA"+K$+"!MRUN!M"

```

```

238PRINT"/" f0: To store data"/" f1: To abort
it."
240*FX15,1
242PROCOSCLI(K$)
244REPEATUNTILGET$="#"
246END
248
250DEFPROCLIST
252RESTORE1000
253ONERROR PROCRESTART
254READA$
256FORJ=0TO1STEP0
257PRINT/"*** "FNLTOD(FNLINE)" ***"
258FORI=0TO9
262PRINTTAB(7)": "A$
264READA$
266NEXT
267A=INKEY(800)
268NEXT
270
272DEFPROCSTART
274PRINTTAB(0,24)/"PRESS ANY KEY TO CONTINUE:"I
NKEY(2000)
276RUN
278
280DEFPROCFOUND
281INPUTLINE"/" FIND : "S$
282RESTORE1000
283ONERROR PROCRESTART
284REPEAT
285REPEAT
286REPEAT
288READA$
290UNTILLEN A$>=LEN S$
292UNTILINSTR(A$,S$)<>0
294L=FNLINE
296RESTORE L
298PROCDISPLAY
300PROCINREC
302UNTILFALSE
304

```

```

306DEF PROCREMOVE
307PRINT"REMOVE : "
308L=FNENTRY
310PROCOSCLI("K.0#" +STR$L+";MRUN;M")
312PRINT"/" f0: TO REMOVE"/" f1: TO ABORT"
314REPEATUNTILGET$="# "
316END
318
320DEFPROCSAVE
322INPUT"/" FILE NAME : "F$
324PROCOSCLI("SAVE"+FNQUO(F$)+STR$~PAGE+" "+STR
$~TOP)
326ENDPROC
328
330DEFPROCFINISH
332PRINT"/" FINISH "/" DO YOU WANT TO SAVE
THIS DATA ?"
334REPEAT$=GET$
336UNTIL$="Y"ORA$="N"
338IFA$="Y"THENPROCSAVE
340END
1000REM START OF DATA
1001DATA"***** *","* DATA PRO
G *","*****",,,,,,
2231DATA"PROGRAM ","DATAPROG","FINISHED","AT 3.0
0 PM",,"BY C.N.GRAHAM",,,
32767DATA"***** END OF DATA *****"
>

```

CHARACTER EDITOR

This is quite a simple character editor but has one important improvement over a lot of other simple character editors. That is this one enables you to edit blocks of characters. So you can create either the normal space-invader (just in one character) or a big monster (2 by 3) character in a single go.

To edit a character after you type RUN the program asks for the size of the block of characters (1,1 if you want to edit a single character). The arrow keys are used for moving the cursor around the block, the space bar sets pixels and the return key deletes them. When you have finished press the escape key then type in the ascii code for the first character of the block. You will then be asked for a line number, this is the line number for the "VDU23, ..." statements when they are merged into your programs. The program ends by spooling the required statements to tape or disk under the file name "G CHAR". This can then be merged into your program using "*exec G CHAR".


```

10
20REM Character editor
30REM For single charecters or blocks of them
up to 2 by 3
40
50MODE3
60
70REM Blue black striped backround
80
90VDU19,0,4;0;
100CLS
110
120REM Characters for grid
130
140VDU23,254,255,129,129,129,129,129,129,255
150VDU23,255,255,255,255,255,255,255,255
160PRINT"" CHARACTER GENERATOR"""""" OBJ
ECT SIZE IN CHAR. MAX {2,3} ";
170INPUTXS,YS
180NPX=XS*8
190NPY=YS*8
200
210REM Output array & character pattern array
220DIMA$(XS,YS),P(NPX,NPY)
230MODE2
240
250REM If you press escape once to often then t
ype GOTO CONT
260
270CONT=230
280
290REM Draw up character grid of the correct si
ze
300
310PROCGRID(NPX,NPY)
320
330REM Exit proc scan when escape key is presse
d
340
350ONERRORGOTO380
360PROCSCAN(NPX,NPY)

```

```

370
380REM On error comes here
390
400NERROROFF
410IFERR<>17THENREPORT:PRINT" at line "ERL:END
420PRINTTAB(0,NPY+1)
430
440REM Yellow on black
450
460COLOUR128
470COLOUR3
480INPUT"ASCII CODE : "CO
490
500REM Calculate the codes for the characters
510
520FOR Y=0TONPY-1
530N=0
540FOR X=0TONPX-1
550N=N-2^(NPX-1-X)*P(X,Y)
560NEXT
570FOR I=0TOXS-1
580A$(I,Y DIV8)=A$(I,Y DIV8)+"," +STR$(255 AND (
N DIV(256^(XS-1-I))))
590NEXT
600NEXT
610MODE3
620VDU19,0,4;0;
630CLS
640INPUT"LINE NO. FOR DEF."NO
650
660REM Out put to current filing system in ASCII
I format
670REM So that it can be merged into othe progr
ans using *exec G_CHAR
680
690PRINT"/"/"/"SPOOLING TO FILE ""G_CHAR""
700*spool G_CHAR
710PRINT"AUTO"STR$NO",1";CHR$13;
720N=0
730FOR Y=0TOYS-1
740FOR X=0TOXS-1

```

```

750PRINT"VDU23,"STR$(CO+N);A$(X,Y)
760N=N+1
770NEXT
780NEXT
790VDU27
800*SPOOL
810END
820
830DEFPROCGRID(NPX,NPY)
840LOCALX,Y
850COLOUR130
860CLS
870
880REM Print vertically in cyan on black
890
900COLOUR128
910COLOUR6
920$="Press Escape To Exit"
930PRINTTAB(18,0);
940FORHZ=1TOLEND$
950VDUASCHID$(D$,HZ,1),8,10
960NEXT
970VDU30
980COLOUR129
990COLOUR0
1000FORY=1TONPY
1010FORX=1TONPX
1020VDU254
1030NEXT
1040PRINT
1050NEXT
1060COLOUR3
1070ENDPROC
1080
1090DEFPROCSCAN(MX,MY)
1100LOCALX,Y
1110REPEAT
1120
1130REM Test cursor keys & move cursor
1140
1150IFINKEY(-58)THEN Y=(Y+MY-1)MODMY

```

```

1160IFINKEY(-42)THEN Y=(Y+1)MODMY
1170IFINKEY(-26)THEN X=(X+MX-1)MODMX
1180IFINKEY(-122)THEN X=(X+1)MODMX
1190COLOUR3
1200PRINTTAB(X,Y);CHR$(254-P(X,Y))CHR$8;
1210
1220REM Test space bar & return key
1230
1240IFINKEY(-99)THENP(X,Y)=TRUE
1250IFINKEY(-74)THENP(X,Y)=FALSE
1260TIME=0
1270REPEATUNTILTIME>10
1280COLOUR0
1290PRINTTAB(X,Y);CHR$(254-P(X,Y))CHR$8;
1300UNTILFALSE
1310ENDPROC
1320

```

>

SPLIT

This is a utility program to expand a program written with long multi-statement lines into a program containing more readable single statement lines. If the new version has enough memory to run it will still work as before expansion but will be a bit slower. It still works because the utility does not split up lines after "IF", "REM" or "DATA" statements. This program was used to make my programs originally written for the PET or TRS-80 more readable after they had been converted to work on the BBC computer.

When the program is going to be used to split the lines of another program, the computer has to be put into mode 7 and PAGE set to &7800 before the program is loaded. If you are using a disk system type "*tape" before running the program. When it is RUN the program loads the first program to be found on tape into memory and proceeds to work on it. As it works it will print on the screen the line numbers of all new lines created. When split has finished you can see the expanded version by setting PAGE to &EOO.

If the program to be expanded has lines containing many statements then it may help to renumber it with line spacings of 20 or so, before saving it to tape to be read by split. Split is written with multi-statement lines so that it uses the minimum amount of memory but this means you can test split on itself before you waste time with longer programs. Note that split can be used on programs as long as the Mutant adventure game before you will run out of memory.

MAIN ROUTINES

Main Program: Loads first file to be found on tape to address &1800. Set pointers to the beginning of old and new files. Initialise first few bytes of the new program. Then calls PROCLINE for each line of the old program.

PROCLINE: Split up one line of the old program.

PROCTRANS: Transfer one byte from old to new program unless it is a token for "IF", "REM" or "DATA" in which case PROCRL is used to transfer all the remainder of the line.

PROCRL: Transfer remainder of line from old to new program.

PROCNEWL: Create new line in expanded version.

PROCQU: Transfer quotes and program between them in one go. So that ":" inside quotes are split.

PROCLINUM: Transfer line number.

```

5*L."" 1800
7P1=&E00:P2=&1800
8?P1=13:P1=P1+1:P2=P2+1
10REPEATPROCLINE:UNTIL?P2=255OR P1+10>P2
15?P1=255
20END
100DEFPROCLINE
110!P1=!P2:NUM=?P2*256+P2?1
120L=P2?2
130I2=3:I1=3
140REPEATPROCTrans:UNTIL I2=L
145P1?2=I1
150P2=P2+L:P1=P1+I1
154PRINTNUM
155I1=3:I2=3
160ENDPROC
170DEFPROCTrans
180IFP2?I2=ASC": "THENPROCNEWL:ENDPROC
190IFP2?I2=&E7 ORP2?I2=&F4 ORP2?I2=&DC THENPROC
RL:ENDPROC
200IFP2?I2=ASC""""THENPROCQU:ENDPROC
210IFP2?I2=&BD THENPROCLINUM
220P1?I1=P2?I2:I1=I1+1:I2=I2+1
230ENDPROC
1000DEFPROCRL
1010P1?2=I1+LEN$(P2+I2)+1:$(P1+I1)=$(P2+I2)
1020I1=I1+LEN$(P2+I2)+1:I2=L
1030ENDPROC
1040DEFPROCNEWL
1050NUM=NUM+1
1055PRINTNUM
1060P1?2=I1+1
1070P1?I1=13
1080P1=P1+I1+1:I1=3:I2=I2+1
1090?P1=NUM DIV256
1100P1?1=NUM AND255
1110ENDPROC
1120DEFPROCQU
1125P1?I1=P2?I2:I1=I1+1:I2=I2+1
1130REPEATP1?I1=P2?I2:IFP2?I2=13THENPRINT"ERROR"
,NUM:STOP

```

```
1135I1=I1+1:I2=I2+1:UNTIL?(P2+I2-1)=ASC""""  
1140ENDPROC  
1150DEFPROCCLINUM  
1160FORK=1TO3:P1?I1=P2?I2:I1=I1+1:I2=I2+1:NEXT  
1170ENDPROC
```

>

CURVE FITTING

One problem when doing high resolution displays is the amount of data required if you want something to have smooth curves. This program, given the co-ordinates of some points on the required curve will try to calculate the value of the constants needed to make a curve of the formula.

$$f(i)=a*i^4+b*i^3+c*i^2+d*i+e$$

Pass through the points given, so that you only need five numbers, that is, the values the program calculates for a,b,c,d,e. So in your program you can draw your curve using:

```
a = . . . . (numbers calculated)
b = . . .
c = . . . .
d = . . .
e = . . .
DEF FNF(i) = a*i4 + b*i3 + c*i2 + d*i + e
k = (Start Position)
MOVE K, FNF(K)
FOR I = K TO (End Position)
DRAW I, FNF(I)
NEXT I
```

You can have vertical curves using FNF as the horizontal component. If you have a very complicated curve it may be necessary to split it into a few simple curves that over-lap.

The program works by first calculating five equations for the difference between your data points and the curve being calculated. These are stored in the array MAT. These are then solved to minimise the total of the differences squared.

The program also tries to find the roots of the curve, that is, the values of "i" for which $f(i) = 0$. You may want these points as the beginning and the end points of a curve.

This program was used for the snake in the program "MUMMY".

Results least squares fit to : $AX^4 + BX^3 + CX^2 + DX + E = Y$
 & roots of the curve.

Number of data points : 12 Average difference : 0.20091989

The coefficients are :

A = -2.00317952E-3

B = 4.63846517E-2

C = -0.307771367

D = 1.47132074

E = 0.89143157

Roots of curve :

-0.540587299

X Given	Y Given	Y Calc.	Diff.
1	2	2.09936241	9.93624125E-2
2	3	2.94201392	-5.79860834E-2
3	4	3.62557953	-0.374420467
4	4	4.3081764	0.3081764
5	5	5.09984534	9.98453423E-2
6	6	6.06255089	6.25508931E-2
7	7	7.21018125	0.210181247
8	9	8.50854833	-0.491451666
9	10	9.87538773	-0.124612268
10	11	11.1803587	0.180358727
11	12	12.2450443	0.245044321
12	13	12.8429511	-0.157048859

```

10MODEO
20PRINT'"/TAB(10);"Least squares fitting progr
am."/
30NZ=5:PRINT"Curve used :  $Y = AX^4 + BX^3 + CX^2 + DX + E$ "
40INPUT" Number of data points : "NP
50DIMX(NP),Y(NP),MAT(NZ,NZ),CONST(NZ),RES(NZ),
ROOT(4),SXP(8)
60PROCINP :REM Input data
70PROCCALC_MAT :REM Convert problem to simulta
neous equations in MAT( )
80PROCCALC_CONST
90PROC GAUSS(5) :REM solve equations stored in
matrix using gaussian elimination
100PROCNEWTON_RAP :REM Find roots of curve
110PROCOUTPUT
120END
130
140DEFPROCINP
150LOCALI
160PRINT'" Please type in data points. If you
make a mistake you will be"
170PRINT"given a chance to correct it at the en
d."/
180FORI=1TONP
190PROCASK_POINT
200NEXT
210IFFNYN("ALL CORRECT")THENENDPROC
220REPEAT
230INPUT" Point number of error",I
240PROCASK_POINT
250UNTILFNYN("ALL CORRECT")
260ENDPROC
270
280DEFPROCASK_POINT
290PRINT"X & Y values of point [";I;"] "":INPUT
X(I),Y(I)
300ENDPROC
310
320DEFFNYN(A$)
330PRINT'A$;
340INPUT" {YES or NO} ".A$

```

```

350IFA$=""THEN340
360=LEFT$(A$,1)="Y"
370
380DEFPROCCALC_MAT
390LOCALI,J
400FORI=1TONP
410FORJ=1TO8
420SXP(J)=SXP(J)+X(I)^J
430NEXT
440NEXT
450SXP(0)=NP
460FORI=1TO5
470FORJ=1TO5
480MAT(I,J)=SXP(10-I-J)
490NEXT
500NEXT
510ENDPROC
520
530DEFPROCCALC_CONST
540LOCALI,J
550FORI=1TONP
560FORJ=1TO5
570CONST(J)=CONST(J)+X(I)^(5-J)*Y(I)
580NEXT
590NEXT
600ENDPROC
610
620
630
640DEFPROC GAUSS(NZ)
650LOCALKZ,LZ,IZ,JZ,SUM,TEMP,FACT
660REM - - - -
670REM Requires array MAT(NZ,NZ),CONST(NZ),RES(
NZ)
680REM - - - -
690FORKZ=1TONZ-1:LZ=KZ
700FORIZ=KZ+1TONZ
710IFABSMAT(IZ,KZ)>ABSMAT(LZ,KZ)THENLZ=IZ
720NEXT
730IFLZ<>KZTHENPROC SWOP(LZ,KZ)
740FORIZ=KZ+1TONZ
750FACT=MAT(IZ,KZ)/MAT(KZ,KZ)

```

```

760FOR JZ=KZ+1 TONZ
770MAT(IZ,JZ)=MAT(IZ,JZ)-FACT*MAT(KZ,JZ)
780NEXT
790CONST(IZ)=CONST(IZ)-FACT*CONST(KZ)
800NEXT
810NEXT
820REM Back solution
830RES(NZ)=CONST(NZ)/MAT(NZ,NZ)
840IZ=NZ-1
850REPEAT
860SUM=0
870FOR JZ=IZ+1 TONZ
880SUM=SUM+MAT(IZ,JZ)*RES(JZ)
890NEXT
900RES(IZ)=(CONST(IZ)-SUM)/MAT(IZ,IZ)
910IZ=IZ-1
920UNTIL IZ<1
930ENDPROC
940
950DEFPROC SWOP(L,K)
960LOCAL J
970FOR J=K TONZ
980TEMP=MAT(L,J)
990MAT(L,J)=MAT(K,J)
1000MAT(K,J)=TEMP
1010NEXT
1020TEMP=CONST(L)
1030CONST(L)=CONST(K)
1040CONST(K)=TEMP
1050ENDPROC
1060
1070
1080DEFPROC NEWTON_RAP
1090LOCAL X, LASTSOL, LASTY, Y, I, DX
1100PRINT "''"===== Newton_Ramphson Equ
ation solving in X=====
1110NSOL=0
1120PROC CREATE
1130INPUT "Minimum value of X",XS"Maximum value o
f X",XE
1140DX=(XE-XS)/50
1150LASTX=XS:X=XS:LASTY=EVAL(A$):LASTSOL=1.7E38

```

```

1160FORI=1TO50
1170X=XS+I*DX
1180Y=EVAL(A$)
1190IFY*LASTY<0THENPROCSOLV
1200LASTY=Y
1210LASTX=X
1220NEXT
1230IFLASTSOL=1.7E38 THENPRINT"No roots found."
1240ENDPROC
1250
1260DEFFNDIFF(FUN$,X)
1270LOCALA,DX
1280DX=X/40000
1290A=EVAL(FUN$)
1300X=X+DX
1310=(EVAL(FUN$)-A)/DX
1320
1330DEFFNN_R(EQU$,X)
1340LOCALY,X1,L
1350L=2*EVAL(EQU$)
1360REPEAT
1370Y=EVAL(EQU$)
1380X1=X
1390X=X-Y/FNDIFF(EQU$,X)
1400IFABS(Y)>ABS(L)THENPRINT"STRING$(40,"*+");"
Not converging X="X;" " ";EQU$;"=";Y:=1.7E38
1410UNTILABS(X-X1)<X/10000OR ABS(Y-EVAL(EQU$))<Y
/10000ORABS(Y)<1E10
1420=X
1430
1440DEFPROC SOLV
1450LOCALRES
1460RES=FNN_R(A$,(X+LASTX)/2)
1470IF RES=1.7E38 THEN ENDPROC
1480IFRES<>LASTSOL THENNSOL=NSOL+1;ROOT(NSOL)=RE
S:LASTSOL=RES
1490ENDPROC
1500
1510DEFPROCGRAPH
1520LOCALMIN,MAX,X,V,VSCALE,HSCALE,Y,VZERO
1530MIN=1E38;MAX=-1E38
1540FORX=XS TOXE STEP(XE-XS)/100

```

```

1550V=EVAL(A$)
1560IFV>MAX THENMAX=V
1570IFV<MIN THENMIN=V
1580NEXT
1590FORI=1TONP
1600IFY(I)>MAX THENMAX=Y(I)
1610IFY(I)<MIN THENMIN=Y(I)
1620NEXT
1630VDU3:CLS
1640VSCALE=959/(MAX-MIN):VZERO=-MIN*VSCALE
1650HSCALE=1279/(XE-XS):X=XS:Y=EVAL(A$)
1660VDU5:FORIZ=1TONP
1670MOVE(X(IZ)-XS)*HSCALE-8,(Y(IZ)-MIN)*VSCALE+1
6
1680VDU120
1690NEXT:VDU4
1700IFVZERO>0ANDVZERO<1280THENMOVE0,VZERO:DRAW12
79,VZERO
1710X=-XS*HSCALE:IFX>=0ORX<1280THENMOVEX,1023:DR
AWX,0
1720MOVE0,(Y-MIN)*VSCALE
1730FORX=XS TOXE STEP(XE-XS)/512
1740DRAW(X-XS)*HSCALE,(EVAL(A$)-MIN)*VSCALE
1750NEXT
1760PRINTGET$
1770CLS:ENDPROC
1780
1790DEFPROC CREATE
1800LOCAL I
1810A$=""
1820FORI=1TO5
1830A$=A$+STR$(RES(I))+"*X^"+STR$(5-I)
1840IFI<>5THENA$=A$+"+"
1850NEXT
1860ENDPROC
1870
1880DEFPROC OUTPUT
1890LOCAL I,J,LP,DIF,TDIFF
1900LP=FNYN("Do you require printer out put")
1910IFLP THENVDU2
1920PRINT"" Results least squares fit to : AX

```

```

^4 + BX^3 + CX^2 +DX + E =Y"
1930PRINT"    & roots of the curve."
1940PRINT/"Number of data points : "NP
1950PRINT/"The coefficients are : "
1960FORI=1TO5
1970PRINTCHR$(64+I);" =",RES(I)
1980NEXT
1990PRINTSTRING$(4,"+-----");"+"
2000PRINT"! X Given";TAB(18);"! Y Given";TAB(36)
; "! Y Calc.";TAB(54);"! Diff."TAB(72);"! "
2010FORI=1TONP
2020PRINT"! ";X(I);TAB(18);"! ";Y(I);TAB(36);
2030X=X(I):Y=EVAL(A$):DIF=Y-Y(I):TDIFF=TDIFF+ABS
(DIF)
2040PRINT"! ";Y;TAB(54);"! ";DIF;TAB(72);"! "
2050NEXT
2060PRINTSTRING$(4,"+-----");"+"
2070PRINT/"Average difference : "TDIFF/NP
2080IFNSOL=0THENPRINT"Curve has no roots."ELSEPR
INT/"    Roots of curve : ":FORI=1TONSOL:PRINT" "RO
OT(I);:NEXT
2090VDU3
2100IFFNYN("Do you require a graph :")THENPROCGR
APH
2110ENDPROC
2120
>

```

MR. UGLY

This program is a demonstration of drawing quite a complex object without having loads of data statements. It draws a face (not quite human) in shades of blue, from black to white. The program draws in character resolution in mode 1 as a hi-resolution display would take a very long time to do. The program scans the screen and uses FNV to calculate the shade of blue for each character position.


```

10REM*****
20REM*      Mr. Ugly      *
30REM*****
40MODE1
50
60REM Redefine colours 1,2 & 3 to blue,cyan &
white
70
80VDU19,1,4;0;19,2,6;0;19,3,7;0;
90R=1.2
100
110REM Loop for each character position in the
picture
120
130FORYZ=0TO30
140FORXZ=5TO34
150COLOURFNV/R*4+128
160PRINTTAB(XZ,YZ)" ";
170NEXT
180NEXT
190END
200
210
220DEF FNV
230
240REM Main shape of the head
250
260H=((XZ-20)*.06)^4
270H=H+((YZ-10)*.072)^2
280H1=((XZ-20)*.075)^4
290H1=H1+((YZ-24)*.12)^4
300
310REM Eyes
320
330E=((XZ-16)*.25)^4
340E=E+((YZ-13)*.45)^4
350E1=((XZ-24)*.25)^4
360E1=E1+((YZ-13)*.45)^4
364
365REM Mouth
366

```

```

370M=((XZ-20)*.11)^4
380M=M+((YZ-24)*.5)^4
384
385REM Nose
386
390N=((XZ-20)*.5)^4
400N=N+((YZ-16)*.3)^4
404
405REM Total up all to produce face
406
410T=1/(H+1)^4+.9/(H1+1)^4
420T=T-1/(E+1)^4-1/(E1+1)^4
430T=T-1/(M+1)^4
440T=T+.4/(N+1)^4
450IFT>0 THEN=T ELSE=0

```

This is a modified version of Mr. Ugly that does its display in 80 columns using different characters to show different shades of gray. With one change marked in the program it can be used to draw the face on a printer, as shown below.

[illegible]

```

10REM*****
20REM*      Mr. Ugly  80 COL. VERSION  *
30REM*****
40MODE0
50
70
72DIMC$(8)
74FORI=0TO8
75READC$(I)
76NEXT
90R=1.2
100
110REM Loop for each character position in the
picture
120
130FORYZ=0TO30
140FORXZ=0TO78
145REM For output to printer change (FNV to (8-
FNV
160PRINTC$(FNV(XZ/2,YZ)/R*9);
170NEXT
175PRINT
180NEXT
190END
200
210
220DEF FNV(XZ,YZ)
230
240REM Main shape of the head
250
260H=((XZ-20)*.06)^4
270H=H+((YZ-10)*.072)^2
280H1=((XZ-20)*.075)^4
290H1=H1+((YZ-24)*.12)^4
300
310REM Eyes
320
330E=((XZ-16)*.25)^4
340E=E+((YZ-13)*.45)^4

```

```

350E1=((XZ-24)*.25)^4
360E1=E1+((YZ-13)*.45)^4
364
365REM Mouth
366
370M=((XZ-20)*.11)^4
380M=M+((YZ-24)*.5)^4
384
385REM Nose
386
390N=((XZ-20)*.5)^4
400N=N+((YZ-16)*.3)^4
404
405REM Total up all to produce face
406
410T=1/(H+1)^4+.9/(H1+1)^4
420T=T-1/(E+1)^4-1/(E1+1)^4
430T=T-1/(M+1)^4
440T=T+.4/(N+1)^4
450IFT>0 THEN=T ELSE=0
1000DATA" ", ., ":", =, ?, P, Q, N, M
>

```

SKETCHING

This is a mode 2 sketching program that uses the right joy-stick to control its graphics cursor. Keyboard input is used to control graphics plotting modes and the fire button is used to make the current line, point, triangle or cursor position permanent, as the program gives a constant display of what the current graphics display would look like if made permanent.

The program will accept the following keyboard commands:

M: Move graphics cursor.

P: Plot a point.

D: Draw line from last cursor position to current position.

Set last position to new end of line.

L: Draw line from last cursor position to current position.

Does not change last cursor position.

T: Draw triangle from last two positions to current position.

@: Print a character typed at cursor position, utilising the "M" command when finished with the character.

C: New colour type a letter "A" – "O" for next colour.

E: End program.

```

2REMSKETCHING
3REM M = MOVE
4REM D = DRAW FROM LAST POS
5REM L = LINE FROM LAST POS. FROM OTHER OPTIO
NS
6REM T = TRIANGLE FROM LAST TWO POINTS
7REM C = CHANGE COLOUR TYPE "C" THEN "A" TO "
0" GIVE YOU THE 16 COLOURS
8REM @ = PRINT CHAR AT CURSOR TYPE SPACE THEN
"@ THEN YOUR CHAR
10REM Sketching program
20MODE2
30PROCINIT
40REPEAT
45REM Flash on screen what current option look
s like
50PROC
55REM If fire button then make permanent
60IFADVAL0AND1THENPROCP
70REPEATUNTIL0=(ADVAL0AND1)
80PROCSTICK
90UNTIL0$="E"
100END
110
120DEFPROCSTICK
125REM Read joy stick to move around the scre
en
130X=X+(2048-ADVAL1DIV16)/200
140X=X+(1280ANDX<0)-(1280ANDX>1279)
150Y=Y+(ADVAL2DIV16-2048)/200
160Y=Y AND1023
170ENDPROC
180
190DEFPROCT
195REM Choose plot option number
200IF0$=0L$THEN260
205PO=FALSE
210IF0$="M"OR0$="E"THENPO=4
220IF0$="L"OR0$="D"THENPO=5
230IF0$="P"THENPO=69
240IF0$="C"THENPROCCC:GOTO260

```

```

250IF0$="T"THENP0=85
252IF0$="Q"THENC=GET:P0=4
2530L$=0$
255REM Test for new option command
260A=INKEY(0):IFA<>-1THEND$=CHR$A
270IFP0=FALSE THENENDPROC
280FORIX=0TO1
290MOVEX2,Y2:MOVEX1,Y1
300PLOTPO,X,Y
310IFTEMP THENPROCCU
320NEXT
330ENDPROC
340
350DEFPROCLOCAL TEMP
360GCOL0,COL
370IF0$<>"Q"THENPROCT ELSEPROCCHAR
380GCOL3,COL
385REM Update old possitions
390IF0$<>"L"THENX2=X1:X1=X:Y2=Y1:Y1=Y
400ENDPROC
410
420DEFPROCCU
425REM Put a "x" at cursor position
430IFCOL=0THENGCOL3,7
440MOVEX-32,Y+16
450VDU5,120,4
455GCOL3,COL
460ENDPROC
470
480DEFPROCINIT
485REM Move to initial position etc
490X=640:Y=512:COL=1:MOVEX,Y
500X1=X:X2=X:Y2=Y:Y1=Y:TEMP=TRUE
510GCOL3,COL
5200$="M":0L$=""
530ENDPROC
540
550DEFPROCCC
555REM Change colour
560REPEAT
570A=INKEY(3000)

```


580UNTILA>64ANDA<83
590COL=A-65
6000\$="M":0L\$="":6COL3,COL
610ENDPROC
620
630DEFPROCCHAR
640MOVEX-32,Y+16
650VDU5,C,4
660ENDPROC
670

>

EFFECTS PROGRAM

This is a battle between a robot from a UFO and the well armed people of a town. It is used to demonstrate some of the special effects that can be done on the BBC computer. The program makes good use of the ENVELOPE command, synchronizing sound channels, the VDU19 for redefining graphics colours. It also uses the right colour combinations and the GCOL options to make an object appear to move behind others.

POINTS OF INTEREST

In line 170, colour 3 is redefined as black, so that in the four colour graphics mode there are two colours defined as black. This is done so that one colour black can be flashed to other colours, independently of the other. This is also needed so that the UFO can fly behind things like the telephone pole.

At line 830 the ENVELOPE used for the robot's footsteps is redefined for each step, with different values for the volume.

ADVAL-5 and ADVAL-6 are used to find the number of free spaces in the sound buffers. This is needed to find out when the last sound finished, so that the next sound effect is in phase with the graphics.

All of the buildings and most of the telephone poles are drawn, using PROCB. This procedure fills in a block, given diagonally opposite corners.

For short delays the INKEY(n) is used and this is useful in debugging the program, as the program can be speeded up a little by pressing keys.

In lines 900-910 the robot is drawn in black (colour 3) on black (colour 0). This is so that during the battle, colour 0 can be flashed to other colours leaving the shadow of the robot visible.

```

10REM*****
20REM*      EFFECTS      *
30REM*****
40
50REM Define characters for UFO.
60
70VDU23,224,0,7,127,255,255,255,17,32
80VDU23,225,0,224,254,255,255,255,136,4
90
100REM Draw a town at dusk.
110
120ENVELOPE1,1,0,0,0,1,1,1,126,-12,-7,-5,126,50
130MODE1
140
150REM Define colour 3 as black.
160
170VDU19,3;0;0;
180
190REM Draw the sky at dusk.
200
210GCOLOR,1
220X=0
230REPEAT
240D=RND(200)
250PROCB(X,X+D,1023,400+RND(200))
260X=X+D
270UNTILX>1279
280
290REM Draw buildings in foreground.
300
310GCOLOR,3
320PROCB(0,250,0,650)
330PROCB(1050,1279,0,700)
340
350REM Draw telephone pole.
360
370PROCB(630,638,0,800)
380FORI=720TO780STEP20
390GCOLOR,3
400PROCB(610,660,I,I-9)
410GCOLOR,0
420MOVE0,I*1.2

```

```

430DRAW635,I
440DRAW1279,I
450NEXT
460
470REM Random lines at the bottom of the screen
.
480
490GCOLOR,3
500FORI=0TO40
510K=RND(200)*RND(1)*RND(1)
520MOVERND(800)+200,K
530PLOT1,RND(50),0
540NEXT
550
560REM UFO lands
570
580ENVELOPE2,140,-1,-1,-1,10,10,10,5,0,-1,-1,12
6,126
590ENVELOPE3,1,-1,-12,3,3,1,10,5,0,-1,-1,100,10
0
600SOUND&101,2,90,50
610SOUND&102,3,90,200
620GCOLOR,3
630FORI=0TO650STEP6
640X=SIN(I/280)*700
650MOVEX,1060-I
660VDU5,224,225,4
670A=INKEY(2)
680MOVEX,1060-I
690VDU5,224,225,4
700NEXT
710
720REM First round of the battle.
730
740FORJ=0TO20
750PROCRF
760TIME=RND(50)
770REPEATUNTILTIME>100
780NEXT
790
800REM Robot foot steps.

```

```

810
820FORV=.1 TO1 STEP.1
830ENVELOPE4,1,0,0,0,1,1,1,100,-4,-1,-7,126*V,1
20*V
840SOUND&100,4,7,40
850SOUND&101,0,100,40
860NEXT
870
880REM Draw robot (in black on black).
890
900GCOLOR,3
910PROCRB
920REPEATUNTILADVAL-5=15
930A=INKEY(42)
940FORJ=0TO20
950PROCRF
960TIME=RND(50)
970REPEATUNTILTIME>100
980NEXT
990
1000REM Robot fires back.
1010
1020ENVELOPE5,132,1,1,1,50,50,50,1,0,0,-126,126,
126
1030ENVELOPE6,1,-8,12,-8,1,1,1,1,0,0,-126,100,10
0
1040
1050SOUND&101,5,20,100
1060SOUND&102,6,20,100
1070MOVE740,375
1080GCOLOR,2
1090VDU5,42,4
1100ENVELOPE7,128,5,3,1,1,1,1,126,-1,0,-5,126,11
0
1110SOUND&101,0,1,1
1120SOUND&100,7,4,30
1130REPEATUNTILADVAL-6=15
1140FORI=0TO190
1150VDU19;RND(15),0;0;
1160NEXT
1170VDU19;0;0;0

```

```

1180GCOL0,0
1190MOVE740,375
1200VDU5,42,4
1210TIME=0
1220REPEATUNTILTIME>400
1230PROCRB
1240
1250REM Robot walks off.
1260
1270FORV=1 TO.1STEP-.1
1280ENVELOPE4,1,0,0,0,1,1,1,100,-4,-1,-7,126*V,1
20*V
1290SOUND&100,4,7,40
1300SOUND&101,0,100,40
1310NEXT
1320REPEATUNTILADVAL-5=15
1330ENVELOPE2,140,1,1,1,20,20,20,5,0,-1,-1,126,1
26
1340TIME=0
1350REPEATUNTILTIME>200
1360
1370REM UFO takes off.
1380
1390SOUND&101,2,90,50
1400SOUND&102,3,90,200
1410GCOL3,3
1420COLOUR129
1430PRINTTAB(0,30);
1440C=0
1450FORI=650TO0STEP-6
1460X=SIN(I/280)*700
1470MOVEX,1060-I
1480VDU5,224,225,4
1490A=INKEY(2)
1500MOVEX,1060-I
1510VDU5,224,225,4
1520C=C+1
1530IFC AND 1THENVDU11
1540NEXT
1550MODE7
1560END

```

```

1570
1580
1590
1600DEFPROC RB
1610
1620REM Robot
1630
1640PROCB(720,740,425,400)
1650PROCB(728,732,400,396)
1660PROCB(708,752,394,388)
1670PROCB(720,740,388,350)
1680PROCB(720,726,350,320)
1690PROCB(736,740,350,320)
1700PROCB(708,712,388,350)
1710PROCB(748,752,388,370)
1720ENDPROC
1730
1740
1750DEFPROC B(X1,X2,Y1,Y2)
1760MOVEX1,Y1
1770MOVEX1,Y2
1780PLOT85,X2,Y1
1790PLOT85,X2,Y2
1800ENDPROC
1810
1820
1830DEFPROC RF
1840
1850REM Random firing.
1860
1870IF RND(2)=1 THEN VDU19;7;0;0;:A=INKEY(4):SOUND0
,1,5,5:VDU19;0;0;0;0;:ENDPROC
1880FOR I=0 TO RND(12)
1890SOUND0,1,6,1
1900GCOL3,2
1910MOVE420,100
1920DRAW570+RND(50),300+RND(30)
1930DRAW420,100
1940NEXT
1950ENDPROC

```

>

MUMMY

This is a graphics demo that draws a pyramid, a mummy and a snake in a desert. It shows the use of the triangle filling routine to fill complex shapes. To draw the mummy the program uses data statements to define the outline of the more complex parts of its body and individual routines for the arms and legs etc. The snake is done in a different way. One function inside a loop is used to define its outline. The constants for that function were calculated by the curve fitting program also in this book. All the shapes are filled in by drawing triangles from one side of the outline to the other and back stepping down the shape as it goes from side to side.

MAIN ROUTINES

Main Program: Draws background and then calls the procedures to draw the objects.

PROCARM: This is used to draw the arms of the mummy.

PROCLEG: Used for drawing the mummy's legs.

PROCSHADOW: Draws a shadow under the mummy.

PROCDOOR: Draws a door in the pyramid.

FNS: This gives outline of the snake. The values of the constants calculated by the curve fitting program.

PROCPYRAMID: Draws pyramid using triangles and dotted lines.

PROCESCAPE: After a short delay this draws an open door in the pyramid and tracks coming out.

PROCMUMMY: Draws mummy.

PROCSNAKE: Draws snake.


```

10REM MUMMY
20MODE2
25
30REM Eye character
35
40VDU23,224,0,112,188,120;0;0
45
50REM Sky
55
60COLOUR132
70CLS
75
80REM Ground
87
90MOVE0,600
100GCOL0,1
110PLOT85,1279,0
120PLOT85,1279,600
124
125REM Texture ground
126
130GCOL0,3
140FORXZ=0TO1279
150FORIZ=1TO2
160PLOT69,XZ,601-RND(600)*RND(1)
170NEXT
180NEXT
184
185REM Background complete
186
190PROCOPYRAMID
200PROCESCAPE
210PROCShadow
220PROCmummy
230PROCsnake
240END
245
250
255
260DEFPROCARM(X,Y)
270MOVEX,Y

```

```

280MOVEX+52,Y
290FORI=1TO250STEP16
300PLOT0+(0=5),X+I/20,Y-I
310PLOT0,52+X-I/20,Y-I
320NEXT
330ENDPROC
340
350DEFPROCLEG(X,Y)
360MOVEX,Y
370MOVEX+62,Y
380FORI=1TO290STEP16
390PLOT0+(0=5),X+I/40,Y-I
400PLOT0,62+X-I/40,Y-I
410NEXT
420ENDPROC
430
440DEFPROCShadow
450GCOLOR,0
460FORIX=0TO200
470PLOT69,800+RND(100+IX),100+RND(250)
480NEXT
490ENDPROC
500
510DEFPROCDOOR(X,Y)
520MOVEX,Y
530PLOT0,16,4
540PLOT81,-20,16
550PLOT81,16,4
560ENDPROC
570
580DEFFNS(T)
590REM Coefficients calculated by the least squ
are fitting program
600T=-6.58456137E-3*T^4+0.267833056*T^3-3.62909
294*T^2+16.9152128*T+6.62503517E-2
610=T*6
620
630REM Data for mummys head & sholders
640DATA2,5,7,8,8.5,7.5,7,6,5,5,7
650DATA15,17,18,19,19
660
670DEFPROCPYRAMID

```

```

680REM Pyramid
690MOVE670,550
700GCOLOR,7
710MOVE440,500
720PLOT85,500,810
730PLOT85,400,580
740GCOLOR,0
750PLOT21,500,810
760PLOT21,440,500
770PLOT21,670,550
780PLOT21,500,810
790MOVE440,500
800PLOT21,400,580
810ENDPROC
820
830DEFPROC ESCAPE
840REM Delay
850FOR JZ=0 TO 20000
860NEXT
870GCOLOR,0
880PROC DOOR(570,520)
890REM Mummy tracks
900FOR JZ=0 TO 150 STEP 2
910PLOT 69,570+JZ*2.5+RND(SQRJZ)*3,520-(JZ/9)^2
920NEXT
930ENDPROC
940
950DEFPROC MUMMY
960REM Mummy
970MOVE900,900
980FOR O=85 TO 5 STEP -80
990MOVE900,900
1000GCOLOR,7 AND (O=85)
1010RESTORE
1020FOR IZ=0 TO 15
1030READX
1040PLOT O+(O=5),900-X*8,900-16*IZ
1050PLOT O,900+X*8,900-16*IZ
1060NEXT
1070FOR IZ=16 TO 30
1080PLOT O+(O=5),822+IZ/6,900-16*IZ

```

```

1090PLOT0,988-IZ/6,900-16*IZ
1100NEXT
1110PROCARM(752,664)
1120PROCARM(1000,664)
1130PROCLEG(828,440)
1140PROCLEG(920,440)
1150GCOLOR,13
1160MOVE860,850
1170VDU5,224
1180MOVE900,850
1190VDU224,4
1200NEXT
1210ENDPROC
1220
1230DEFPROC SNAKE
1240REM Snake
1250MOVE300,550
1260MOVE300,550
1270GCOLOR,2
1280FORI=0TO16STEP.5
1290PLOT85,300-FNS(I),550-I*34.375
1300PLOT85,300+FNS(I+.25),550-(I+.25)*34.375
1310NEXT
1320GCOLOR,0
1330FORX=-120TO120STEP4
1340PLOT69,300+X,400+X^2/300
1350NEXT
1360FORX=-64TO64STEP4
1370PLOT69,300+X,420+X^2/200
1380NEXT
1390FORIX=-1TO1STEP2
1400MOVE300,420
1410GCOLOR,10
1420DRAW300+8*IX,380
1430GCOLOR,0
1440PLOT69,300+IX*10,430
1450MOVE284+IX*60,500
1460GCOLOR,15
1470VDU5,224,4
1480NEXT
1490ENDPROC

```

>

3D GRAPH

This short program draws a three dimensional surface as defined by the function at line 330. The program calculates which points on the surface are hidden by other parts of the surface and plots accordingly.

When it is RUN the program puts the screen in mode 0. Then working from the left edge to the centre of the screen, the surface is scanned from front to back. For each point scanned its value of FNA is calculated, the height as seen on the screen as if the surface was tilted 30 degrees towards you, if this is higher or lower than all the points in front of it then plot it. The mirror image is also plotted on the other side of the screen.

Lines of interest:

110: Calculates edge of area to be plotted.

140: Calculates vertical height of point to be plotted.

330: Function being plotted.

```

10 REM 3D Graph in mode 0
20 REM == =====
30 MODE0
40 REM Work from both edges towards it's centr
e
50 FORX=-150TO0
60   K=5
70   L=0
80   U=3E37
90   P=1
100  Z1=0
110  Y1=K*INT(SQR(22500-X*X))/K)
120  FORY=Y1 TO-Y1 STEP-K
130    REM Calculate height of each point from
a view point 30 deg. above horizontal.
140    Z=INT(80+FNA(SQR(X*X+Y*Y))-.5*Y)
150    REM Is it invisible part of the top of
the surface ?
160    IFZ<L THEN230
170    M=1
180    L=Z
190    PLOT69,640+M*X*4,Z*4+100
200    IFM=1THEN M=-1:GOTO190
210    GOTO280
220    REM Jump if not visiable part of bottom
surface
230    IFZ>U THEN280
240    M=1
250    U=Z
260    PLOT69,640+M*X*4,Z*4+100
270    IFM=1THEN M=-1:GOTO260
280    NEXT
290  NEXT
300 END
310
320 REM Function being plotted
330 DEF FNA(Z)=38*(SIN(Z/20)+.48*SIN(3*Z/12))+2
0
>

```

RECURSIVE PATTERN

This program starts by drawing a straight line. Then it draws another further up the screen but this time the middle third of it is replaced by two lines of equal length to the missing section, joined to the line and each other at 60 degrees. This is then repeated for two more lines with each section having its middle replaced as above. The result is that each line is a more complex shape than the one below.

POINTS OF INTEREST

Lines 20-55 sets the starting points for each line.

Line 60 draws each line.

PROCREC is a recursive procedure (it calls itself). Its parameters are used as follows:

A: Angle in radians in which to start drawing line.

D: Total length of line to be drawn.

L%: Is the depth of recursion.

The variable L% controls the behaviour of PROCREC: If L% is zero then only a straight line is drawn in direction A and the procedure is exited. Otherwise PROCREC is called four times to draw a line with the centre substituted as above.

```

5REM Recursive pattern
10MODE0
20FORLX=0TO3: REM Draw 4 lines.
30X=64*(3-LX): REM Start position
40Y=8+200*LX: REM for each line.
50MOVEX,Y
55REM Draw line of level LX.
60PROCRC(0,1280-128*(3-LX),LX)
70NEXT
80END
90
100DEFPROCRC(A,D,LX)
105REM If level is 0 then draw line.
110IFLX=0THENPROCnew:DRAWX,Y:ENDPROC
120PROCRC(A,D DIV3,LX-1)
130PROCRC(A+RAD60,D DIV3,LX-1)
140PROCRC(A-RAD60,D DIV3,LX-1)
150PROCRC(A,D DIV3,LX-1)
160ENDPROC
170
175REM Calculate new values of X & Y.
180DEFPROCnew
190X=X+D*COSA
200Y=Y+D*SINA
210ENDPROC

```


KEY BUFFER

When this demonstration program is RUN it prompts the user with "+" instead of the normal ">". Basic commands like PRINT and LIST will still work if they are less than 20 characters long.

The program prints its own prompt ("+") and inputs a line of text. It then uses the operating system call called OSBYTE (&FFF4) number 138 to put characters of the text in the keyboard buffer, as if they had just been typed in. It then puts a RETURN character (13) and "RUN" RETURN in the buffer and ends the program. So after the program has finished, the basic interpreter is left, with your command, in the keyboard buffer. Once this has been completed there is still the "RUN" and return character in the keyboard buffer, so that the program is started again.

This type of routine can be used by a program to add data statements to itself, for example: If the program requires the values of A, B, C and D to be a permanent part of the program then the statements:

```
PROCIN LINE("10000DATA"+STR$A+"", "STR$B+", +STR$C+"", "+  
STR$D)
```

```
PROCIN LINE ("GOTO100")
```

```
END
```

would add the data statement at line 10000 and restart the program at line 100.

```

1REM { OS 1.0 ONWARDS }
5REM Demo of FX 138
6REM Takes a line of text from user
7REM then puts it in the key board
8REM buffer followed by a "RETURN"
10INPUTLINE"+ "A$
15IFLENA$>26THENPRINT"TOO LONG":GOTO10
16REM (A$="RUN" would cause program
17REM to get stuck in a loop.)
20IFA$<>"RUN"THENPROC_IN_LINE(A$)
25REM Put "RUN" &"RETURN" in buffer
30PROC_IN_LINE("RUN")
35REM End prog so that key board
36REM buffer is used.
40END
50
60DEFPROC_IN_LINE(L$)
70L$=L$+CHR$(13):L=LENL$
80FORIX=1TOL
90XZ=0:YZ=ASC MID$(L$,IX,1)
95REM YZ=char to be placed.
100AZ=138 :REM FX call number.
110CALL&FFF4 :REM Put char in buff.
120NEXT
130ENDPROC

```

HILBERTS CURVES

This program draws a series of patterns known as Hilberts Curves, each one takes a full screen to display so they are displayed one at a time and the screen cleared in between each one.

The program consists of a main program whose main purpose is to set the starting point on the display for each call of PROCP. The main routine calls PROCP once to draw each pattern with the second parameter increased by one each time. This parameter is the maximum recursion level (number of times that PROCP can recursively call itself before it does a ENDPROC). The first parameter is the starting direction for drawing the pattern.

```

10 REM__HILBERTS CURVES
20 N=7:H0=950
30 MODE0
40 PRINT"HILBERT CURVES"
50 I=0:H=H0:X0=H DIV 2:Y0=X0
60 REPEAT I=I+1:H=H DIV2
70   X0=X0+H DIV2:Y0=Y0+H DIV 2
80   X=100+H0:Y=Y0+10
85   MOVEX,Y
90   PROCP(1,I)
95   A=GET:CLS
100  UNTIL I=N
110 END
115
120 DEFPROCP(TYP,I)
130 IF I<=0 THEN ENDPROC
140 ON TYP GOTO 150,160,170,180
150 PROCP(4,I-1):X=X-H:PROCM
151 PROCP(1,I-1):Y=Y-H:PROCM
152 PROCP(1,I-1):X=X+H:PROCM
153 PROCP(2,I-1):ENDPROC
160 PROCP(3,I-1):Y=Y+H:PROCM
161 PROCP(2,I-1):X=X+H:PROCM
162 PROCP(2,I-1):Y=Y-H:PROCM
163 PROCP(1,I-1):ENDPROC
170 PROCP(2,I-1):X=X+H:PROCM
171 PROCP(3,I-1):Y=Y+H:PROCM
172 PROCP(3,I-1):X=X-H:PROCM
173 PROCP(4,I-1):ENDPROC
180 PROCP(1,I-1):Y=Y-H:PROCM
181 PROCP(4,I-1):X=X-H:PROCM
182 PROCP(4,I-1):Y=Y+H:PROCM
183 PROCP(3,I-1):ENDPROC
190 DEFPROCM
200 DRAWX,Y
210 ENDPROC

```

INPUT 0-9

This is another demo of the operating system call called OSWORD this time number 0. OSWORD zero is the main part of the basic statement INPUT. This program uses it to input a string from the keyboard 20 characters long and ignoring all characters not digits 0-9. The ability to ignore characters is more useful when it is used to ignore control and tab characters.

For more information on operating system calls see the User Guide.

```

10REM Demo of selective keyboard
20REM input using OSWORD (0)
30REM In this case
40REM ( Only accept 20 digits. )
50MODE7
60PRINT'""CHR$131"IGNORE CHAR OUTSIDE ""0""-"
"9""
70PRINTCHR$129"PLEASE TYPE IN SOME RUBBISH"
80OSWORDZ=&FFF1
90REM Dimension control block &
100REM input buffer
110DIMSZ 4,BZ 256
120REM Set XZ & YZ to point to
130REM control block XZ (lsb)
140XZ=SZ AND255
150YZ=SZ DIV256
160REM Set pointer to buffer
170!SZ=BZ
180REM Set max number of digits =20
190SZ?2=20
200REM Set range of input char.
210SZ?3=ASC"0"
220SZ?4=ASC"9"
230REM AZ OSWORD CALL NUMBER =0
240AZ=0
250PRINTCHR$133":";
260CALLOSWORDZ
270PRINT"OK"''';VDU130
280REM Show contents of buffer
290PRINT$BZ
300PRINT'""AGAIN (Y)";
310*FX15,1
320REM Flush keyboard buffer
330IFGET$="Y"THEN RUN
340VDU132,136:PRINT" BYE"
>

```

SOUND

This is a demonstration of how to get sound out of your computer the hard way. It uses the operating system call OSWORD (&FFF1) number 7 instead of the SOUND command. I cannot think of any reason why you should need to use this in your basic programs but this is just to show how to use OSWORD calls from basic.

```

10MODE7
20REM Demo sound using OSWORD
30REM Eg. make sound the hard way.
40OSWORDX=&FFF1
50REM Dimension control block
60REM Plus 2 so that "!" can be used
70DIM SZ 10
80PRINT"" SOUND VIA. OSWORD""
90REM Ask Mr. K. Basher
100PRINT" SOUND C,V,P,D"
110VDU131:INPUT"SOUND "CZ,VZ,PZ,DZ
120!SZ=CZ
130SZ!2=VZ
140SZ!4=PZ
150SZ!6=DZ
160REM Set 6502 registers
170AZ=7
180REM X,Y address of control block
190XZ=SZ AND255 :REM lsb
200YZ=SZ DIV256 :REM msb
210REM Make sound
220CALL OSWORDX
230PRINT""AGAIN (Y)";
240REM Flush keyboard buff and ask
250*FX15,1
260IFGET$="Y"THEN 80
270VDU133,136:PRINT"BYE""

```


FOUR PATTERN GENERATORS

These four programs show how easy it is to produce high resolution patterns on the BBC computer. The patterns they produce are easy to change by experimenting with the numbers etc in the programs.

They include demonstrations of the use of SIN, COS and recursive procedures in drawing patterns.

Listing 1

```
10 REM **** PATTERN ****
20 MODE0
30 FORA=-PI TOPI STEPPI/100
40   MOVECOSA*500+600,SINA*500+512
50   DRAWSINA*500+600,COSA*500+512
60   DRAW600-SINA*500,512-COSA*500
70   DRAWCOSA*500+600,SINA*500+512
80   NEXT
90 END
```

Listing 2

```
10REM*****
20REM****TRIANGLES RULE*****
30REM*****
40
45MODE0
50PROCTRIANGLE(640,512,350)
55PROCTRIANGLE(640,512,-350)
60END
70
1000DEFPROCTRIANGLE(X,Y,S)
1005GCOL4,S MOD3+1
1010MOVEX+S,Y
1020MOVEX-S*0.866,Y+S*.5
1030PLOT85,X-S*.866,Y-S*0.5
1035IFABSS<80THENENDPROC
1040PROCTRIANGLE(X+S,Y,S/2)
1050PROCTRIANGLE(X-S*0.866,Y+S*0.5,S/2)
1060PROCTRIANGLE(X-S*0.866,Y-S*0.5,S/2)
1070ENDPROC
```

Listing 3

```
10MODE1
15FORJ=3TO12
17CLS
20FORI=0TO1279STEPJ
30MOVEI,0
40GCOL3,1
50DRAW640,1023
60GCOL3,2
70MOVEI,1023
80DRAW640,0
90NEXT
100NEXT
```

Listing 4

5 REM Graphic pattern using mode 0 & exclusive or (GCOL3,)

```
10 MODE0
15 GCOL3,1
20 FORI=0TO4608STEP4
30 MOVE640,512
40 DRAWFNX(I),FNY(I)
50 NEXT
60 END
999
1000 DEF FNX(P)
1010 P=P MOD 4608
1020 IFP<1280 THEN=P
1030 IFP<2304 THEN=1279
1040 IFP<3584 THEN=3583-P
1050 =0
1060
2000 DEF FNY(P)
2010 P=P MOD 4608
2020 IFP<1280 THEN=0
2030 IFP<2304 THEN=P-1279
2040 IFP<3584 THEN=1023
2050 =4607-P
```

DIGIT CLOCK

This program displays a digital clock on the screen in MODE 3. It keeps time using the TIME function and the main routine to produce the display is machine code.

MAIN ROUTINES

Main program

10-50 Define OS call entry points and reserve space for machine code.

60 Assemble machine code program. If you want the assembly language listing then replace the 2 by a 3.

70 Input time.

80 Error checking: if time is entered in the wrong format or contains a silly time then a Mistake error is produced so that the time is asked again.

90 Call function FNCT with the input as its params. Thus an error will be generated if the input had the wrong format so it will go to line 70 via line 80.

130-170 Display time once a second until a key is pressed.

FNCT 200-220 Check that hours, minutes and seconds are all in the range 0-50 and generate a mistake error if not.

230 Convert time to hundredths of a second.

FNTI Convert time to output format. Reset clock after 24 hours.

FNF Convert number to two digit string.

PROCLARGE Print in large characters.

PROCASSEMBLE Assemble machine code program to print in large characters. Where the parameters X and Y are the position of the display on the screen and N% is the assembler option (OPT N%).

```

1REM DIGITAL CLOCK { OS 1.0 ONWARDS ONLY }
100SWORD=&FFF1
200SASCI=&FFE3
300SWRCH=&FFEE
40MODE7
50DIM B%8,OUT%9,CODE%150
60PROCASSEMBLE(7,10,2)
70INPUTLINE'""TIME (HH,MM,SS) : "A$
800NERRORIFERR<>17THENPRINT"SORRY?":GOTO70 ELS
ESTOP
90TIME=EVAL("FNCT("+A$+"")")
100NERROR OFF
110MODE3
120PRINT'""START TIME : "FNTI;
125
130REPEAT
140TIM%=FNTI
150PROCLARGE(TIM%)
155REM Wait for seconds to change
160REPEATUNTILTIM%<>FNTI
170UNTILNOTINKEY(1)
175
180END
181REM _ _ _ _ _
185
186REM Convert time to 1/100 of sec.
190DEFFNCT(HZ,MZ,SZ)
200IF SZ>59 OR SZ<0 THEN error
210IF MZ>59 OR MZ<0 THEN error
220IF HZ>23 OR HZ<0 THEN error
230=(SZ+MZ*60+HZ*3600)*100
240
245REM Convert time to output format.
250DEFFNTI LOCALHZ,MZ,SZ,TZ
255REM Reset time after 23:59:59
260IFTIME>8640000THENTIME=0
270TZ=TIME
280SZ=TZ DIV 100 MOD 60
290MZ=TZ DIV 6000 MOD 60
300HZ=TZ DIV 360000 MOD 60
310=FNF(HZ)+": "+FNF(MZ)+": "+FNF(SZ)
320

```

```

325REM Format number to 2 digits.
330DEFNF(NZ) =RIGHT$("0"+STR$NZ,2)
340
345REM Poke text to output buffer &
346REM then large print it.
350DEFPROC LARGE(T$)
360$OUTZ=T$
370CALL CODEZ
380ENDPROC
390
395REM Assemble to large print at X,Y
400DEFPROC ASSEMBLE(X,Y,NZ)
410FI=&70
420FJ=&71
430FL=&72
440P2Z=&73
445REM Two passes last 1 OPTion NZ
450FOR IZ=0TONZSTEPNZ
460PZ=CODEZ
470[OPT IZ
480 LDA #31 \TAB(0,Y)
490 JSR OSWRCH
500 LDA #0
510 JSR OSWRCH
520 LDA #Y-1
530 JSR OSWRCH
540 LDA #1 \FOR I=1 TO 7
550 STA FI
560.FD_I
570 LDX #X \TAB(X)
580 LDA #32
590.ST
600 JSR OSASCI
610 DEX
620 BNE ST
630 LDA #0 \FOR J=0 TO 7
640 STA FJ
650.FD_J
660 LDX FJ \?BZ=OUTZ?J
670 LDA OUTZ,X
680 STA BZ

```

```

690 LDX #BZMOD 256
700 LDY #BZDIV 256
710 LDA #10 \REM Get char pattern
720 JSR OSWORD
730 LDA #128 \REM bit mask
740 STA P2%
750 LDA #1
760 STA FL \FOR L=1 TO 7
770.FO_L
780 LSR P2% \P2%=P2%DIV2
790 LDX FI \REM test bit
800 LDA BZ,X
810 AND P2%
820 BEQ BLANK
830 LDA BZ \VDU?BZ
840 BNE CONT
850.BLANK
860 LDA #32 \VDU32 REM" "
870.CONT
880 JSR OSASCI
890 INC FL \NEXT
900 LDA FL
910 CMP #8
920 BNE FO_L
930 INC FJ \NEXT
940 LDA FJ
950 CMP #8
960 BNE FO_J
970 LDA #13 \PRINT
980 JSR OSASCI
990 INC FI \NEXT
1000 LDA FI
1010 CMP #8
1020 BNE FO_I
1030 RTS \REM return to basic
1040]
1050NEXT :REM End of pass
1060ENDPROC

```

OSFILE

This is just a simple demonstration of the operating system load file routine called OSFILE (&FFDD). This operating system routine loads a file at a given address or the file's own load address. It is the main part of the OS *load command. If you always want to load the same file at the same address then * load is easier. For more information see User Guide.

Lines of interest in the program.

30 When the program is RUN it asks for a load address. If your reply is LOA this then gives the program the address of the MODE7 screen (for loading teletext graphics displays).

40 Reserves memory for machine code (more than is required).

70 Load A reg. with 255 Command for load.

80-90 Set X & Y registers to point to control block.

100 JMP OSFILE is equivalent to JSR OSFILE RTS

170 Evaluate load address and store it. Will accept PAGE, LOA or hex or decimal numbers.

190 Input and store file name.

```

10CLS
20OSFILE=&FFDD
30LOA=&8000-1024 :REM Address for loading thin
gs over screen (mode7)
40DIMBUFF 1024
50FORI=0TO3STEP3
60PZ=BUFF
70[OPTI:LDA#255 \set 6502 registers
80LDX# BLOCK MOD256 \address of control block
90LDY# BLOCK DIV 256
100JMP OSFILE \same as jsr osfile rts
110.BLOCK
120J
130NEXT
140NAME=BLOCK+20 :REMspace for file name
150!BLOCK=NAME :REMpointer to name
160INPUT'/"LOAD ADDRESS",A$
170BLOCK!2=EVALA$ :REMstore load addr.
180BLOCK!6=0
190INPUT'/"FILE NAME "$NAME
200CALLBUFF

```

>

FILL DEMO

At the end of this program is the data for a machine code routine to fill in any area on a graphics screen that has a border of a given colour in any given colour. The routine will work in modes 0,1,2,4 & 5. The area it is to fill must be fully enclosed by the border colour otherwise the fill colour will escape and try to fill the rest of the screen. The data for the machine code program is read in from the data statements and checked as it is put into memory by lines 30-100.

To run the demo program PAGE must be set to &1100 and then the program CHAINED from tape. The program then puts the machine code into memory &0E00 to &1100. It then draws a cloud and a tree and uses the fill routine to fill them in.

NOTES ON THE PROGRAM

The parameters of PROCFILL are the x & y coordinates of the start position for the fill, colour in which to fill the area and the border colour of the area.

PROCTREE is a recursive routine and calls itself to draw the branches of the tree.

```

10REM Tape System only
20IFPAGE<>&1100THENPRINT"Error PAGE <> &1100":
END
30CS=0
40FORIX=&E00 TO&102C STEP4
50READA$
60A=EVALA$
70!IX=A
80CS=CS EORA
90NEXT
100IFCS<>&35B1F6C8 THENPRINT"DATA ERROR ":END
110MODE1
120VDU19;4;0;0;0,19,2,2;0;0;0;
125PROCCLLOUD(750,700,170)
130MOVE550,80
140GCOLOR,2
150PI=.15
160M=15
170PROCTREE(540,80,PI/2,150,.7)
180BRAW550,80
190PLOT77,100,250
200PLOT77,1000,250
210PROCFILL(640,90,1,2)
220PROCFILL(40,90,2,2)
230END
240DEF PROCFILL(XZ,YZ,CZ,BZ)
250CALL&E00,XZ,YZ,CZ,BZ
260ENDPROC
270
280DEFPROCTREE(X,Y,D,W,P) LOCALX2,Y2
290IFW<M THEN420
300X2=X+W*2*COSD
310Y2=Y+W*2*SIND
320DRAWX2,Y2
330PROCTREE(X2,Y2,D+RND(1)/10+P,W*2/3,P-P1)
340X2=X2+W/2*COS(D-PI/2)
350Y2=Y2+W/2*SIN(D-PI/2)
360PROCTREE(X2,Y2,D-RND(1)/10-P,W*2/3,P-P1)
370X2=X+W*COS(D-PI/2)
380Y2=Y+W*SIN(D-PI/2)
390DRAWX2,Y2
400ENDPROC

```

```

410
420X2=X+W*5*COS(D-PI/8)
430Y2=Y+W*5*SIN(D-PI/8)
440DRAWX2,Y2
450DRAWX+W*COS(D-PI/2),Y+W*SIN(D-PI/2)
460ENDPROC
470
480DEF PROCCLOUD(X,Y,S)
490GCOLOR,7
500FORI=0TO2*PI STEPPI/5
510X1=X+S*2*SINI
520Y1=Y+S*COSI
530PROCCIRCLE(X1,Y1,S/2)
540NEXT
550GCOLOR,0
560PROCCIRCLE(X,Y,S*.8)
570PROCCIRCLE(X,Y,S*.51)
575PROCFILL(X,Y,3,3)
580ENDPROC
590
600DEF PROCCIRCLE(X,Y,S)
610MOVEX,Y+S
620FORJ=0TO2*PI STEPPI/30
630DRAWX+S*2*SINJ,Y+S*COSJ
640NEXT
650ENDPROC
660
830DATA&F42087A9,&C002A9FF,&A90CF000,&F001C004,
&F004C006,&8508A902,&600AD72
840DATA&74D004C9,&C90603AD,&AD6DD004,&4C90606,&
9AD66D0,&D004C906,&60CAD5F
850DATA&58D004C9,&850601AD,&602AD70,&A07185,&C6
8D70B1,&70B1C80E,&AD0EC78D
860DATA&70850604,&850603AD,&70B18B71,&B1C87685,
&AD778570,&70850607,&850608AD
870DATA&12A98B71,&A9FFEE20,&FFEE2000,&808570B1,
&ADFFEE20,&7085060A,&850608AD
880DATA&8570B171,&1290187F,&69462D00,&70206C6C,
&6D617261,&72657465,&76A50073
890DATA&A50EC88D,&EC98D77,&C90EB820,&6014D0FF,&
EA0C6A2,&F12009A9,&ECAADFF

```

900DATA&2B7F360,&C500009C,&C5E8F07F,&BAE4F080,&
AD0EC58E,&AD480EC6,&AD4B0EC7
910DATA&AD480EC8,&BA480EC9,&F00EC5EC,&778568CA,
&680EC98D,&C88D7685,&C78D680E
920DATA&C68D680E,&F93200E,&788501A9,&19A97985,&
A9FFEE20,&FFEE2045,&200EC6AD
930DATA&C7ADFFEE,&FFEE200E,&EE2076A5,&2077A5FF,
&76A5FFEE,&8D04E938,&77A50EC8
940DATA&C98D00E9,&EB8200E,&CB2079A6,&9079860F,&
FE52003,&691876A5,&EC88D04
950DATA&6977A5,&200EC98D,&78A60EB8,&860FCB20,&2
0039078,&AD380FE5,&72E50EC6
960DATA&B00EC68D,&EC7CE03,&C88D76A5,&8D77A50E,&
B8200EC9,&F0FFC90E,&F07FC50B
970DATA&F080C507,&F0A4C03,&AD0EE74C,&7C850EC6,&
B50EC7AD,&EB8207D,&1CF0FFC9
980DATA&18F07FC5,&14F080C5,&850EC6AD,&EC7AD7C,&
C6EE7D85,&EEE2D00E,&DDD00EC7
990DATA&C68D7CA5,&8D7DA50E,&C9600EC7,&A204D0FF,
&C5601B01,&C5F8F07F,&8AF4F080
1000DATA&900A2,&1B01D038,&69186860,&10088D01,&8D
006968,&E0BA1009,&AD131003
1010DATA&AD480EC6,&AD480EC7,&AD480EC8,&4C480EC9,
&2E000F48,&61657241,&6F6F7420
1020DATA&6D6F6320,&78656C70,&0,&0,&0,&0,&0

INDEX

STAR TREK	7
A real time fight against the Klingons, with nine game levels	
SPACE BLASTER	23
Three dimensional arcade type game. Can you blast them before they blast you?	
BRICK BASHER	34
Breakout with a few surprises for the beginner and the expert.	
SPACE INVADERS	42
What the invaders lack in speed is made up for by their guided bombs.	
ANIMALS	49
A guessing game in which the computer learns more animals from you as you play.	
BATTLE SHIPS	55
This program plays a real mean game of this schoolboy's favourite.	
MUTANT	72
An adventure game played on rather a nasty planet. Re- forms each time you play it.	
ON THE WAY TO THE INTERVIEW	93
An adventure game with text and a map of your local area.	
DATA PROGRAM	106
A very simple in memory filing system.	
CHARACTER EDITOR	114
Create entire blocks of characters in one go.	
SPLIT	119
Utility for splitting all the long lines in a program into shorter ones.	
CURVE FITTING	122
If you want to know the equation of that line going through those dots you plotted.	
MR UGLY	130
A program to draw a face without using DATA statements.	

MR UGLY 80	133
A version of the above for use with a printer.	
SKETCHING	136
Full colour sketching program using the joystick.	
EFFECTS PROGRAM	140
Combined graphics and sound for guns, machine guns, flying saucers, aliens lasers.	
MUMMY	146
Graphics Demo. Someone left the door open on the pyramid.	
3D GRAPH	151
Demo of 3 dimensional surface plotting.	
RECURSIVE PATTERN	153
The computer's working on strange lines.	
KEY BUFFER	155
Demo of placing characters in keyboard buffer.	
HILBERTS CURVES	157
Graphics demo commonly done in Pascal.	
INPUT 0-9	159
Demo of using OSWORD input routine.	
SOUND	161
Demo of sound using OSWORD.	
FOUR PATTERN GENERATORS	163
Showing how very short programs can produce interesting patterns.	
DIGIT CLOCK	165
Digital clock producing large display on screen.	
OSFILE	169
How to load memory from tape or disk using the OSFILE routines.	
FILL DEMO	171
Hi-res fill routine with demo program.	

Duckworth Home Computing

BBC PROGRAMS 1

Edited by Nick Hampshire

This book provides you with a range of useful and exciting programs for the BBC Micro. Games, utilities, graphics and functional programs are covered. The games include an exciting version of Star Trek, a full length adventure game, Space Invaders, Battleships, Space Blaster, Brick Basher, and many others. Among the functional programs is a personal information retrieval package which enables you to create and manipulate up to 365 records. This is a basic book for every user of the BBC Micro.

Written by Carl Graham and edited by Nick Hampshire, publisher of *Commodore Computing International*.

ISBN 0-7156-1734-6



9 780715 617342

Duckworth
The Old Piano Factory
43 Gloucester Crescent, London NW1

ISBN 0 7156 1734 6
IN UK ONLY £6.95 NET